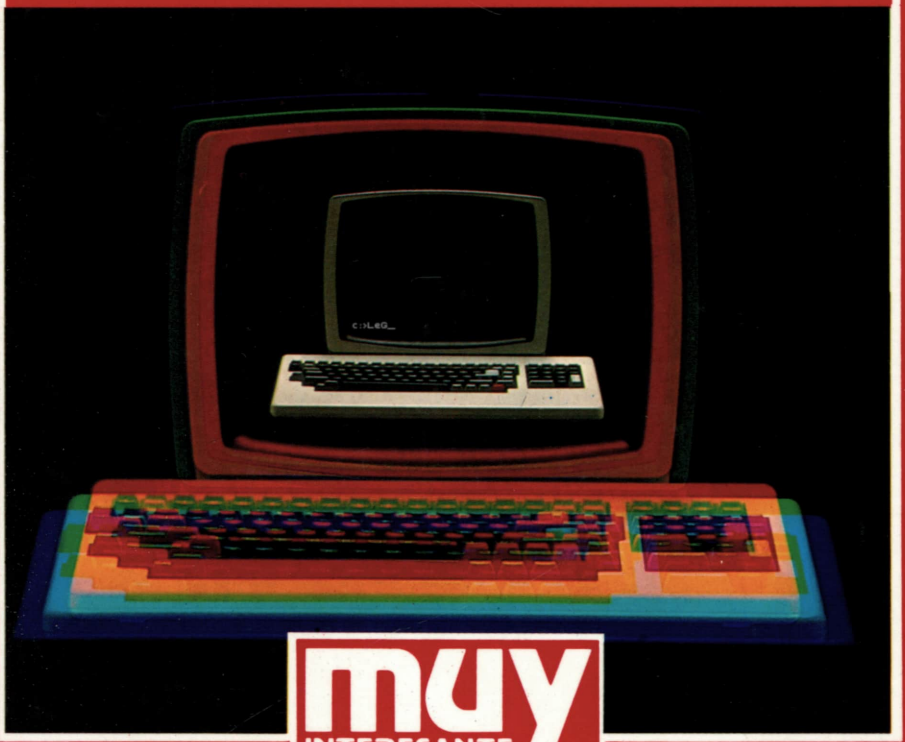
# LIBRO BÁSICO DEL ORDENADOR PERSONAL (I)

1. Los primeros pasos
2. Cómo funcionan

Peter Rodwell



muy INTERESANTE

BIBLIOTECA DE DIVULGACIÓN CIENTÍFICA

*Biblioteca*
*de Divulgación Científica*

# LIBRO BASICO DEL ORDENADOR PERSONAL

# LIBRO BASICO DEL ORDENADOR PERSONAL

PETER RODWELL

I

1. Los primeros pasos
2. Cómo funcionan

Ediciones Orbis, S.A.

**LIBRO BASICO DEL ORDENADOR PERSONAL**

**Título original:**
*The Personal Computer Handbook*
**Publicado por**
Dorling Kindersley Ltd., London

**Project Editor:**
Jonathan Hilton
**Art Editor:**
Neville Graham
**Editor:**
Richard Dawes
**Designer:**
Mark Richards
**Managing Editor:**
Alan Buckingham
**Art Director:**
Stuart Jackman

**Traducción:**
Remigio Gómez
**Asesor científico de la colección:**
Pedro Puigdomènech
**Director editorial:**
Virgilio Ortega

© Dorling Kindersley Ltd.,
London, 1983, 1984.
© Peter Rodwell, 1983, 1984,
del texto original.
© MicroTextos, s.a. de ediciones,
Madrid, 1984 de la 1ª edición
y de la traducción en español.
© Ediciones Orbis, S.A.
Barcelona, 1986
por la presente edición
© por la ilustración de la cubierta:
Don Carroll/The Image Bank

**Nota del editor:**
El *Libro básico del ordenador personal* será publicado en tres volúmenes, que tendrán los números 67, 71 y 75 de esta colección.

ISBN (obra completa): 84-7634-646-8
ISBN (libro 67): 84-7634-647-6
Depósito Legal: NA-699-86 (I)

**Fotocomposición:**
Pérez-Díaz, S.A., Madrid

**Fotomecánica:**
F. E. Burman Ltd., London
Progreso Gráfico, Madrid

**Impresión y encuadernación:**
Gráficas Estella, S.A.
Estella (Navarra), 1986

*Printed in Spain*

# SUMARIO

# SUMARIO VOLUMEN I


INTRODUCCION

1

22 ¿Qué tienen de interés los ordenadores?

2

25 ¿Qué es un ordenador?

3

36 ¿Qué es un sistema informático?

4

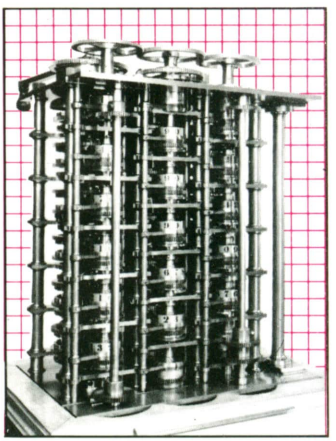
46 Un ordenador por dentro

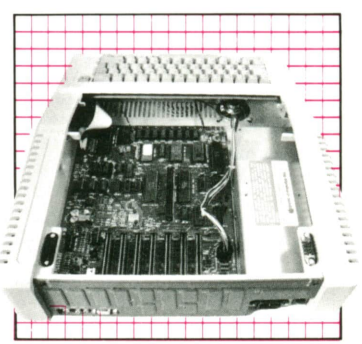
5

54 Aplicaciones de la informática

6

58 Historia de la informática

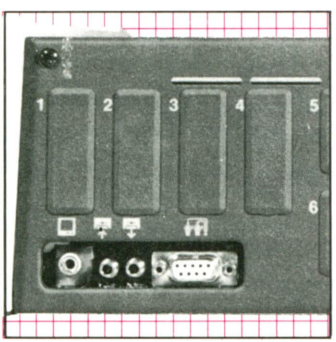





## LOS PRIMEROS PASOS


7

80 La puesta en marcha



8

88 Familiarización

9

96 Los magnetófonos

10

101 Discos y cartuchos

11

105 Consejos sobre programación

12

113 Detección de averías


## COMO FUNCIONAN


13

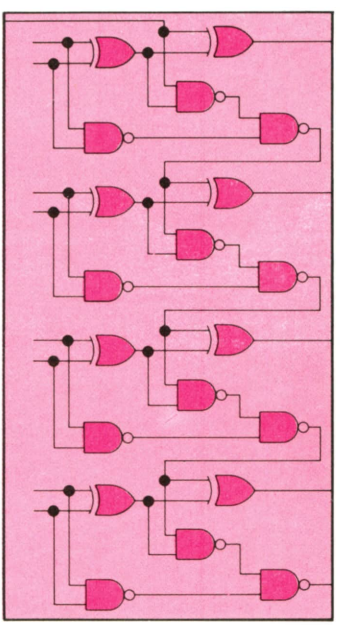
120 Cómo «piensan» los ordenadores

14

129 La lógica informática

15

134 La lógica electrónica


16

144 El ordenador real

# SUMARIO VOLUMEN II

## HARDWARE


17

156 El *hardware* y la CPU

18

166 El chip

19

174 La memoria

20

178 Entrada y salida

21

191 *Cassettes* y cartuchos

22

196 Los discos

23

205 Las impresoras

24

215 Periféricos especiales

## SOFTWARE


25

228 Introducción al *software*

26

239 Los algoritmos

27

246 Programación en Basic

28

272 Depuración de programas

29

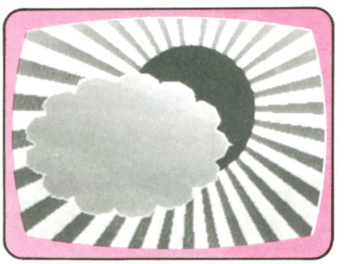
277 Juegos de ordenador

30

282 Informática gráfica

31

287 Programación en ensamblador


32

296 Otros lenguajes de programación

33

315 Software de sistemas

34

326 Bases de datos

35

331 Hojas de cálculo

36

336 Procesamiento de textos

# SUMARIO VOLUMEN III

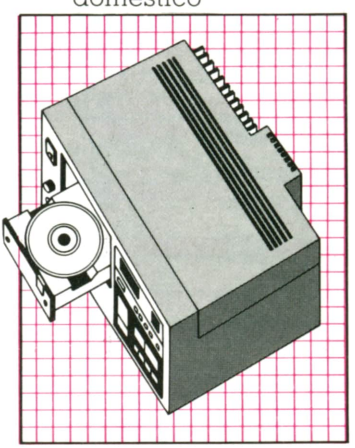
## EL FUTURO


37

344 Inteligencia artificial

38

349 Ordenador doméstico


39

365 Los ordenadores en el trabajo

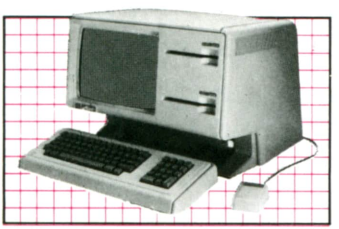
## ELIJA SU ORDENADOR


40

380 Cómo elegir ordenador

41

386 Apple


42

394 IBM

43

398 Sinclair

44

403 Commodore

45

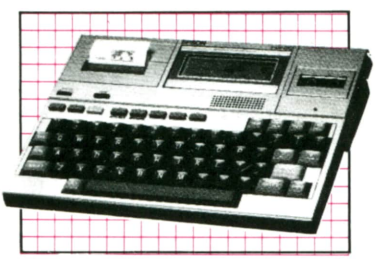
410 Tandy Radio Shack

46

417 Atari

47

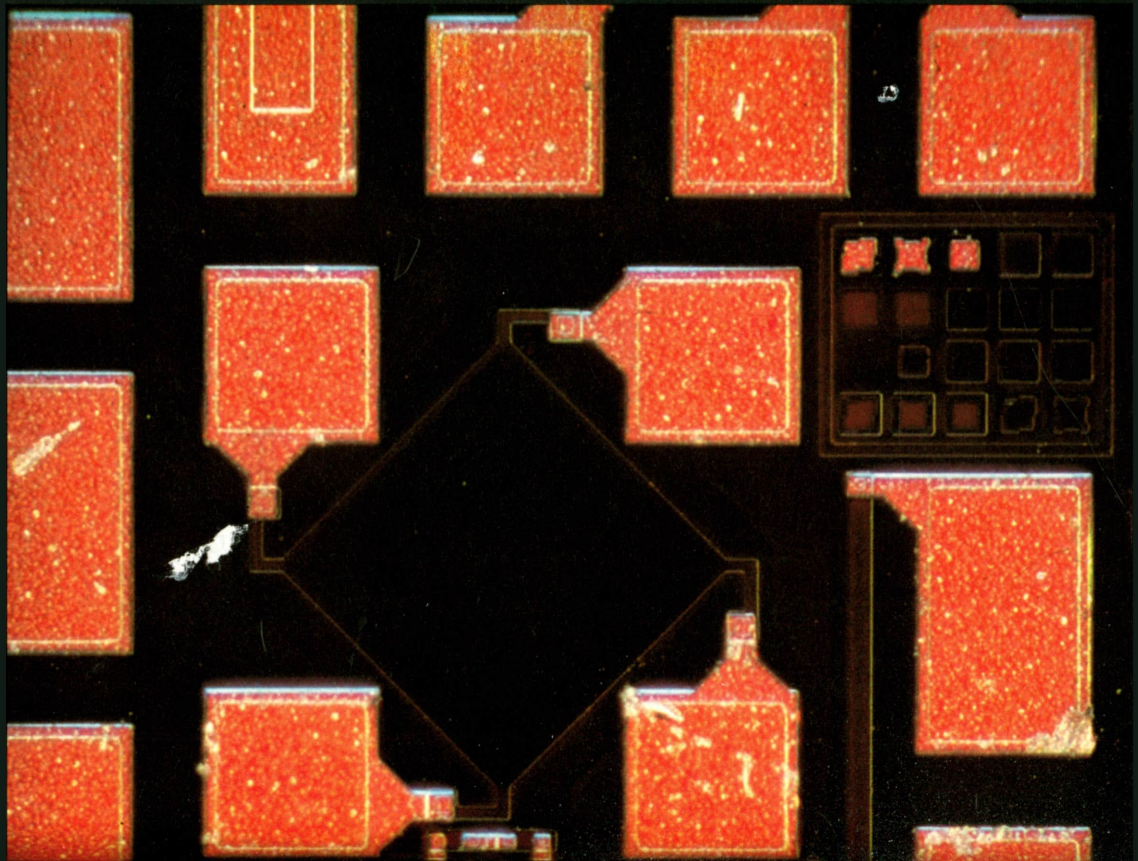
422 Otras máquinas

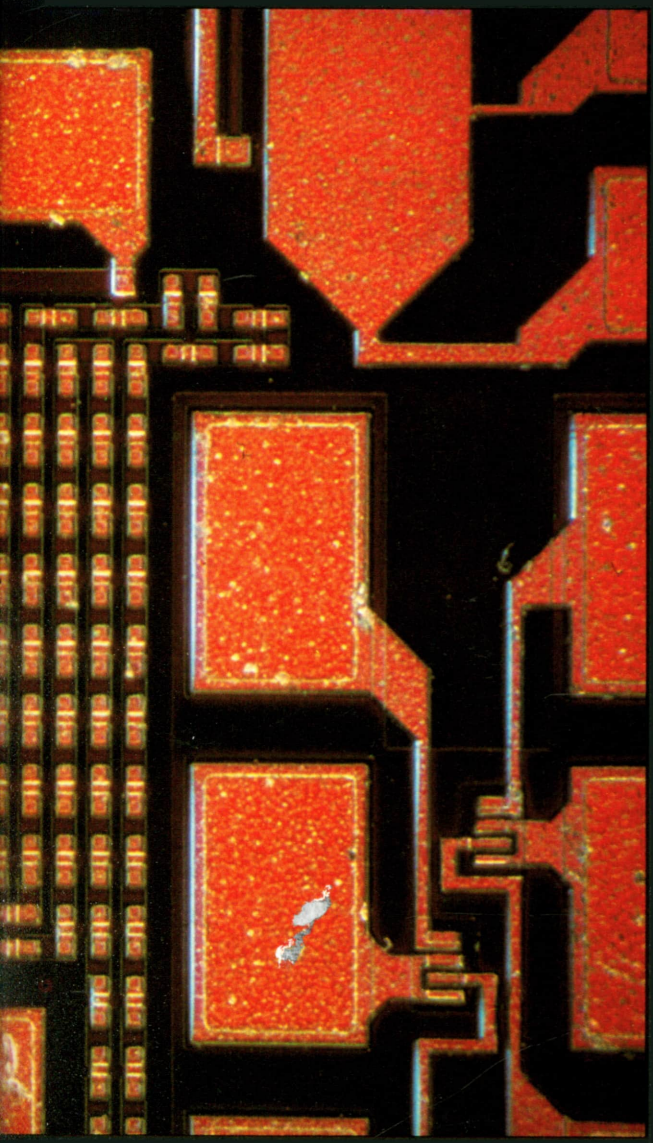




455 Glosario

481 Indice alfabético

494 Agradecimientos





1. Los primeros pasos

2. Cómo funcionan

# INTRODUCCION

Bienvenido al mundo del ordenador. No parece que pase día sin que se anuncie algún nuevo y maravilloso adelanto tecnológico, y en buen número de ellos desempeña algún papel el ordenador. Los objetos cotidianos adquieren de día en día posibilidades novedosas e inesperadas gracias a la tecnología informática. No es de extrañar que nos sintamos algunas veces incómodos por la velocidad y la importancia de tantos avances. Pero aunque en un principio pueda parecernos lo contrario, descubriremos en esta obra que es más una cosa de adaptación del ordenador a nuestro mundo que de nosotros al suyo.

El propósito de este libro es dar al lector una idea de los cambios que va a acarrear la proliferación del uso de los ordenadores en el hogar y el trabajo. Si nos parece interesante esta nueva tecnología, nada mejor que ir a comprarnos uno de los ordenadores personales que son tan baratos ahora. Aunque al principio no usemos más que los juegos prefabricados que se venden, iremos familiarizándonos con la tecnología del ordenador y cobrando conciencia de lo que la constituye.

Tradicionalmente, en la ciencia ficción y en los anuncios publicitarios de la industria informática nos han inculcado la idea de que el ordenador es algo todopoderoso, lo que está muy lejos de responder a la realidad.

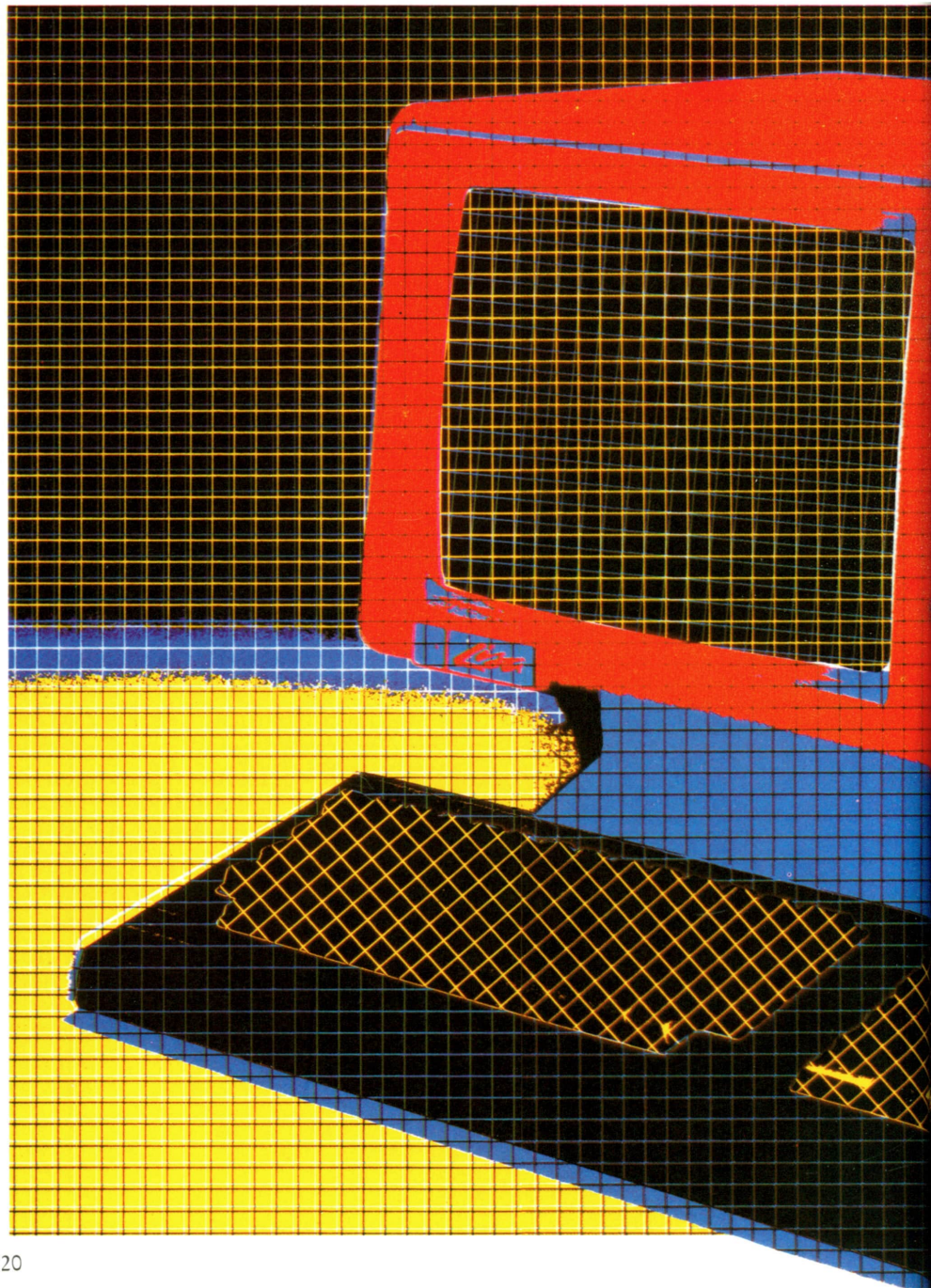
**Cómo usar este libro**

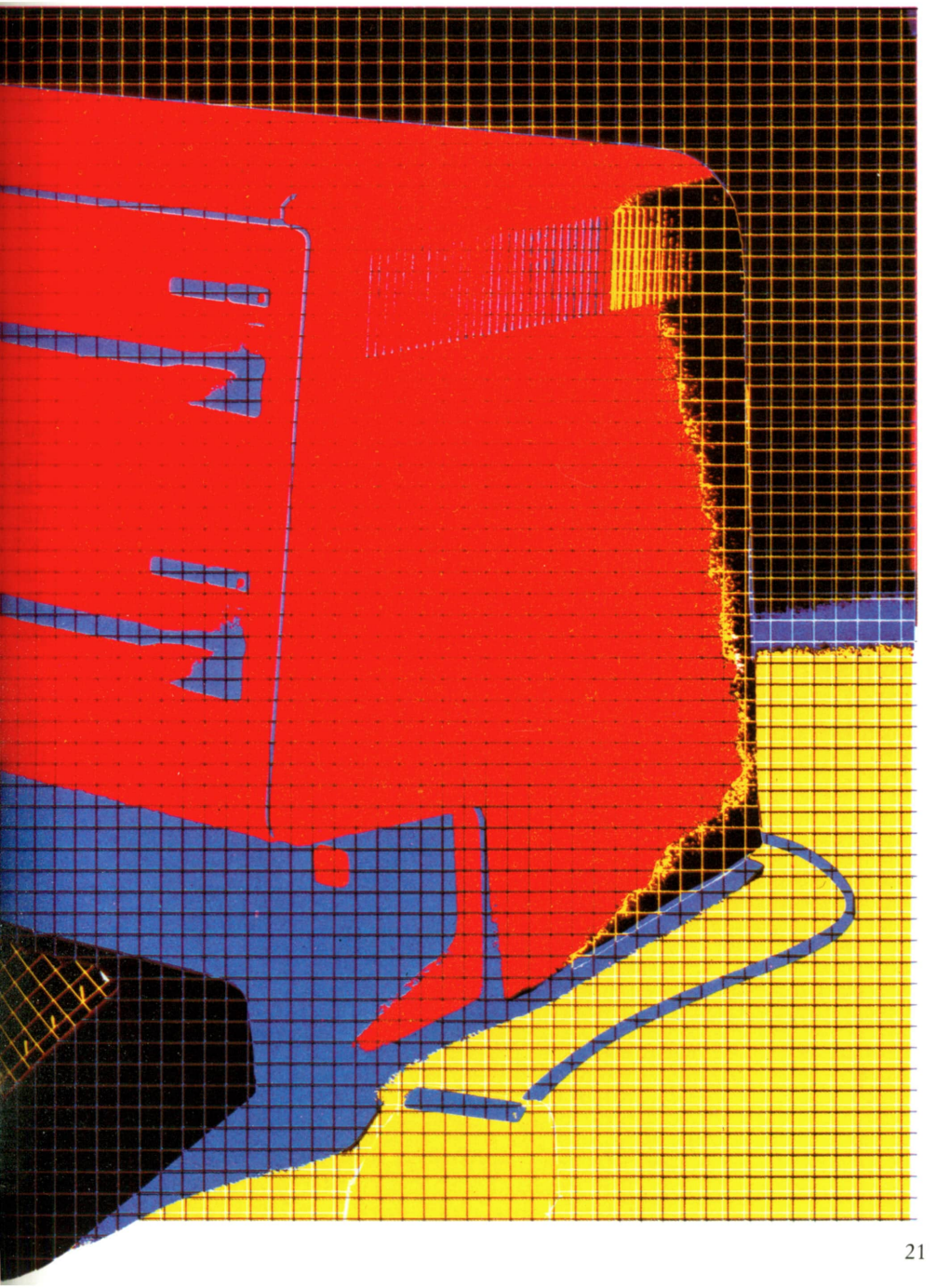
La informática es un tema con multitud de ramificaciones, cada una de las cuales exigiría y exige un libro entero para abordarla en profundidad. Lo que hemos hecho en este libro es dar respuesta a unas cuantas preguntas básicas:

— ¿qué es un ordenador?
— ¿qué sabe hacer un ordenador?
— ¿cómo funciona?
— ¿en qué me afecta a mí, en el hogar y en el trabajo?

Y otras muchas.

El libro posee una estructura global flexible, llevando cada sección un color diferente para que sea fácil distinguirlas. Es posible, si se quiere, leerlo del primer módulo autónomo al último. O se puede ir saltando de uno a otro, captando las corrientes paralelas de saber consultando las numerosas referencias cruzadas que remiten a explicaciones de términos que tal vez no entienda el lector. Por desgracia, el tecnicismo es un mal necesario. Constituye una especie de taquigrafía verbal que nos permite comunicar sucintamente un concepto, sin necesidad de entrar cada momento en prolijas explicaciones. Ya verá el lector que en seguida va cogiendo la terminología, pero si da con alguna expresión o término poco familiar, consulte el glosario, donde está todo explicado.





# ¿QUE TIENEN DE INTERES LOS ORDENADORES?

Hoy día los ordenadores están casi en todas partes, de una manera u otra. Hace años los ordenadores ocupaban enormes salas, tenían un peso de varias toneladas, precisaban de un costoso equipo de aire acondicionado y estaban atendidos por una numerosa plantilla de expertos. Eran monstruos potentes e inaccesibles que exigían que las cosas se hicieran a su modo y hacían estragos cuando se descomponían.

Pero ahora es muy fácil tener algún ordenador en casa incluso sin darnos cuenta. La tecnología ha llegado a fabricar un ordenador no mayor que la uña del dedo meñique, tal vez no muy potente, pero susceptible de ser utilizado en una gama de aplicaciones sorprendentemente amplia. Y allí donde se emplee, aportará más versatilidad o una mayor sencillez de uso y también, en ocasiones, más engorro o incomodidad cuando la tecnología no se aplique o se aplique mal.

Pasando revista a los objetos que tenemos en nuestro hogar podremos comprobar si somos o no usuarios de un ordenador. Fijémonos en el televisor: ¿tiene mando a distancia? Si es así, lo más probable es que lleve dentro un pequeño ordenador. El teléfono, ¿es capaz de recordar números y marcárnoslos al apretar un botón? Aunque no sea más que un teléfono corriente, al usarlo ponemos en marcha una gran cantidad de equipos informati-

zados cuya misión es hacer llegar nuestra llamada a la persona cuyo número hemos marcado.

Por supuesto que también es posible que tengamos en casa un verdadero microordenador: millones de personas poseen ya ordenadores personales, y las ventas de estas máquinas, compactas pero versátiles, se incrementan vertiginosamente. También en el trabajo es cada vez mayor el número de personas que entran en contacto con el ordenador. Desde los comienzos de la informática, las empresas, la ciencia y la enseñanza han sido grandes usuarios de los ordenadores, pero ahora podemos verlos en las mesas de personas que, sin haber estudiado explícitamente informática, se sirven de ellos como útiles de trabajo. Los ordenadores ayudan a los directores a planificar, permiten acceder a la información, facilitan la redacción de cartas, informes y libros. Con tal de elegir adecuadamente el sistema, estas maquinitas pueden potenciar considerablemente no sólo la calidad de nuestro trabajo sino, lo que es más importante, la calidad de nuestra vida laboral, liberándonos de trabajos tediosos y rutinarios.

Ciñéndonos ya al interrogante planteado como cabecera del módulo, no resulta sencillo explicar exactamente qué es lo que hace tan interesante al ordenador, pero no cabe duda de que, una vez superados los prejuicios iniciales, es en verdad fascinante. Y también puede llegar a crear adicción: no es insólito que un amigo nos cuente que se ha pasado media noche en vela tratando de desentrañar los entresijos de determinado programa que le estaba dando guerra.

Parte de la fascinación que el ordenador ejerce reside probablemente en su misma complejidad. Como ya veremos, un ordenador es una compleja colección de elementos básicamente muy sencillos, pero esta complejidad puede crearnos problemas a la hora de tratar de hacer algo con él. Uno de los problemas principales es el de aprender un lenguaje común comprensible tanto por nosotros como por el ordenador. Salvando este

obstáculo, la vía de acercamiento a esta nueva herramienta queda libre.

El interés de los ordenadores se debe también a su enorme difusión, y esta cualidad depende a su vez de otra propiedad importante: cuando lo compramos o entramos en contacto con él por primera vez, el ordenador es una máquina «en blanco». Hay que decirle exactamente, paso a paso, lo que tiene que hacer y, dentro de ciertos límites, se le puede programar para que lleve a cabo una gama increiblemente variada de tareas. Por ejemplo, imaginémonos al ordenador convertido en una máquina de escribir de versatilidad insuperable; quizás, un momento después, decidamos hacer frente a la cruda realidad y solicitarle un resumen de nuestra situación financiera; y luego, tras recobrarnos de la impresión, pensar que lo que nos conviene es sin duda relajarnos, y retarle a una partida de ajedrez; lo más probable es que, salvo que seamos grandes ajedrecistas, perdamos. Estas aplicaciones no son más que algunas de entre las posibles.

No es fácil explicar la fascinación que ejercen los ordenadores. Con la lectura de este libro se hará una idea de lo que dan de sí. Y comprando uno lo averiguará enseguida.

# ¿QUE ES UN ORDENADOR?

Vivimos rodeados de complejos aparatos electrónicos: televisores, vídeos, cadenas de alta fidelidad, lavadoras, hornos de microondas, teléfonos de teclado, etc. Cada día son más los aparatos de esta especie que llevan impresas en lugar destacado las palabras «ordenador», «micro» o «digital». Palabras que han creado su propia mística, pues en el lenguaje de la alta tecnología vienen a significar eficacia, fiabilidad y la calidad de lo más avanzado. Y sin embargo, ¿cuál es la verdadera relación entre estos productos y el ordenador? ¿Qué es exactamente un ordenador?

Todos los aparatos mencionados han sido creados con la idea de que lleven a cabo una tarea específica y sólo una. El ordenador es diferente; es una máquina «en blanco», incapaz de hacer algo por sí misma. Es, esencialmente, una colección de circuitos electrónicos dispuestos de manera tal que pueda responder de forma predecible a cualquier impulso o entrada que reciba. La naturaleza de esa respuesta, sin embargo, puede alterarla a voluntad el usuario. Esta versatilidad es la que distingue al ordenador de los otros aparatos.

A los ordenadores hay que proporcionarles una lista de instrucciones (o «programa») que especifique detalladamente cada uno de los pasos que deben seguir para realizar la tarea que se les va a encomendar. Una vez recibidas, las seguirán

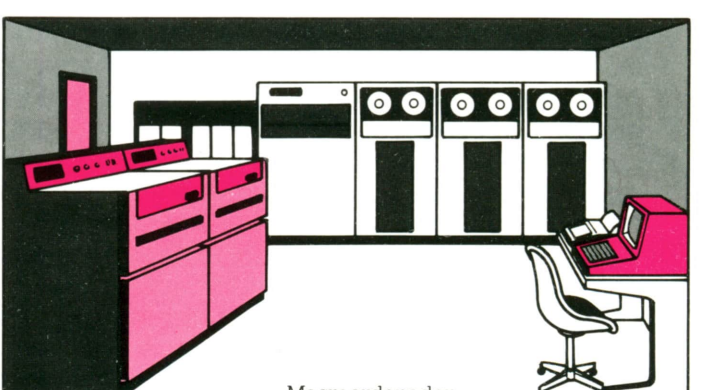
puntualmente, haciendo justo lo que cada una de ellas les indique, ni más ni menos. Los programas son de tal naturaleza que si cometemos un error en las instrucciones el resultado no será el deseado. El ordenador no puede saber lo que *queríamos* que hiciera.

Sin embargo, una vez que hemos conseguido darle el conjunto de instrucciones necesarias para que lleve a cabo una tarea, podemos darle otro conjunto diferente para que haga otra distinta. El ordenador es por tanto un aparato polifacético.

Algunos aparatos, como las lavadoras actuales, llevan a veces un pequeño ordenador. Y en realidad la única diferencia es que el ordenador de la lavadora está dedicado a una sola tarea, controlar la máquina, porque se le ha dotado de un con-

## DEL GRAN ORDENADOR AL MICRO

Hay tres clases de ordenadores. Los macroordenadores, utilizados por grandes empresas, universidades y ministerios. Son de una potencia enorme, y capaces de atender a cientos de usuarios al mismo tiempo. Un miniordenador no es más que un



Macroordenador

junto de instrucciones permanente. Teóricamente sería posible quitarlo, sustituirlo por un conjunto nuevo, y hacerle llevar a cabo una tarea totalmente diferente.

**Los distintos tipos de ordenador**

Los ordenadores suelen clasificarse convencionalmente en función de su tamaño y potencia de cálculo. En el nivel más alto encontramos los macroordenadores *(mainframes)*, gigantes utilizados por grandes empresas, ministerios y centros universitarios y de investigación científica. Los macroordenadores son enormemente potentes, capaces de manejar muchos trabajos distintos al mismo tiempo, permitiendo que cientos de personas vean atendidas simultáneamente sus necesi-

macroordenador en pequeña escala. Tiene capacidad para que varias personas trabajen con él simultáneamente. Su precio es menor, así como las personas necesarias para manejarlo.

Un microordenador es el que utiliza como unidad central de proceso un *chip* de microprocesador, de bajo coste y fabricado en serie. Pocos microordenadores pueden atender a más de un usuario a la vez, pero como son tan baratos es posible darle a cada usuario una máquina.

Miniordenador

Microordenadores

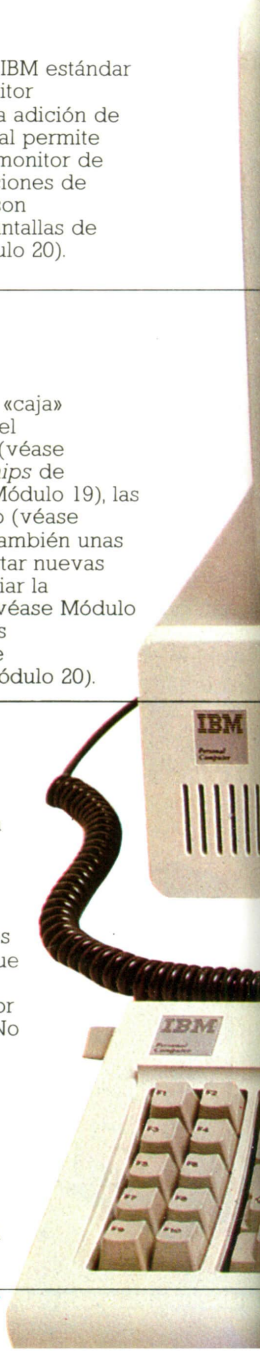
# EL PRIMER MICRO DE IBM

La llegada de IBM al escenario del micro contribuyó a hacer «respetable» la industria a los ojos de muchas personas y estimuló a otros fabricantes de grandes ordenadores a descender a «la parte baja del mercado». El impacto de esta máquina en el negocio de los micros ha sido enorme. Un detalle innovador fue la incorporación de un procesador de 16 bits —uno de los primeros micros en hacerlo—. En Estados Unidos, el reducido precio del modelo básico de la gama le convirtió en un importante competidor en el segmento superior del mercado de ordenadores para el hogar y la diversión, así como en el de pequeños ordenadores de gestión, y en torno a él se ha desarrollado una gigantesca industria que suministra no sólo *software*, sino también tarjetas de expansión directamente incorporables al «PC», como suele llamársele. IBM aguardó dieciocho meses antes de lanzar el PC en Europa, y esta demora permitió que otros sistemas de 16 bits se le adelantaran.

**Monitor.** El PC de IBM estándar viene con un monitor monocromático. La adición de una tarjeta opcional permite sustituirlo por un monitor de color. Para aplicaciones de gestión y juegos son importantes las pantallas de color (véase Módulo 20).

**Unidad central.** La «caja» principal alberga el microprocesador (véase Módulo 17), los *chips* de memoria (véase Módulo 19), las unidades de disco (véase Módulo 22). Hay también unas ranuras para insertar nuevas tarjetas para ampliar la capacidad RAM (véase Módulo 19) y abrir puertos suplementarios de entrada/salida (Módulo 20).

**Teclado.** IBM sabía ya mucho sobre teclados (véase Módulo 20). El teclado del PC es uno de los mejores que existen, aunque haya recibido algunas críticas por ciertos aspectos. No obstante, todo el que esté familiarizado con una máquina de escribir IBM tradicional se sentirá enseguida como en su casa con el teclado del PC.



IBM

dades de cálculo. Pero su precio es elevadísimo, exigen operarios muy cualificados, y salas con aire acondicionado y libres de polvo.

En el nivel inferior en cuanto a potencia de cálculo, está el «miniordenador». Se trata de una versión reducida del gran ordenador, utilizando técnicas de diseño similares. La mayoría de los minis pueden atender a unos pocos usuarios simultáneamente, pero muchos no precisan de la atención de un especialista con plena dedicación ni de unas condiciones ambientales determinadas para funcionar adecuadamente.

Actualmente, no resulta fácil definir con precisión qué es un mini. Por un lado, los minis más potentes superan a los macroordenadores más modestos y, por otro, hay micros que superan a algunos minis. Se ha llegado a decir que un miniordenador no es más que el ordenador fabricado por un fabricante de miniordenadores.

Los días del actual mini están contados. El «microordenador» u «ordenador personal» está ocupando su lugar. Algunos micros ofrecen ya mayor potencia y más prestaciones a precios más bajos que algunos minis pequeños, e incluso desafían también a los minis de la gama media.

Un microordenador es una máquina montada en torno a un microprocesador o «pastilla de silicio» *(silicon chip).* Los micros son tan baratos porque se pueden fabricar en serie, mientras que los minis y los macroordenadores se construyen a mano. Esto, junto a los continuos avances tecnológicos en el diseño y la fabricación de *chips,* es lo que ha posibilitado la revolución de los micros. Cada nueva generación de micros parece ofrecer mayor potencia en un espacio más reducido y con un coste menor.

Como los micros son tan baratos, casi cualquiera puede permitirse el lujo de tener uno. Comprar veinte micros cuesta mucho menos que un miniordenador capaz de atender a igual número de usuarios. Además, para ampliar el sistema basta comprar más micros.

## EL ORDENADOR DE BOLSILLO

El desarrollo de la microelectrónica ha conducido a la invención del microprocesador (véase Módulo 6), y el ordenador de sobremesa no es el único resultado. Hoy día hay ordenadores de tamaño tan reducido que pueden considerarse «de bolsillo». La máquina de la figura, llamada PC1, la fabrica Sharp. Cuenta con un procesador, una corta pero práctica dotación de RAM (véase Módulo 19) y un intérprete Basic completo (véase Módulo 27) en ROM. El PC1 puede usarse como simple calculadora, o escribiendo programas en Basic que se almacenan en RAM. Como opción suplementaria, una *interface* de magnetófono directamente conectable (véase Módulo 21) permite guardar programas en cinta.

**Pantalla de cristal líquido,** de una línea. Puede desplazarse el texto arriba y abajo, a izquierda y derecha, lo que permite visualizar líneas más largas que la pantalla.

**Teclado.** Dispone de un teclado alfanumérico completo (Módulo 20).

**Teclas numéricas.** Se han agrupado estas teclas para su uso como calculadora.

# EL ORDENADOR PORTATIL

Hemos pasado de las salas llenas de equipos de varias toneladas de peso, como eran los ordenadores de hace diez o quince años (véase Módulo 6), a las máquinas portátiles.

Hay dos tipos de portátiles: unas, como la de las páginas 34-35, son versiones reducidas de las máquinas de sobremesa y cuentan con teclados del tamaño habitual, pequeñas pantallas y unidades de disco. Precisan también de alimentación externa. Otras, como la máquina de la derecha, llevan pilas para su total portabilidad.

**El portátil tipo cartera.** Epson es famosa por sus impresoras matriciales (Módulo 23), muy fiables y baratas. Incluso IBM recurrió a Epson para su famoso microordenador PC. Epson hizo su irrupción en el mercado con el QX-10, y el ordenador portátil que muestra la figura, el HX-20. En lugar de hacer una máquina como la de la página siguiente, Epson optó por una configuración mucho más pequeña. La máquina tiene aproximadamente el tamaño de un folio, y cuenta con una pantalla de cristal líquido de cuatro líneas. El teclado, sin embargo, es del tamaño habitual. Lleva incorporada una diminuta impresora matricial y se le puede añadir un microcassette para almacenar datos y programas. Cuenta asimismo con una *interface* RS232 por la que se le puede conectar una pantalla de mayor tamaño.

**Microcassette.** Opcionalmente, puede conectarse a la máquina un microcassette. Servirá para guardar y recuperar programas y datos.

**Pantalla de cristal líquido.** Presenta cuatro líneas simultáneamente, pero una *interface* suplementaria permite conectarle una pantalla de mayor tamaño.

Altavoz

Tecla de alimentación de papel

**Teclado.** El HX-20 cuenta con un teclado como el de las máquinas de escribir, que lo convierte en procesador de textos portátil.

**Impresora.** El ordenador lleva incorporada una diminuta impresora que, aunque demasiado pequeña para el tratamiento de textos, permite poner por escrito tablas y cifras.

**Unidades de disco.** Las unidades de disco gemelas (Módulo 22) vienen insertas en la máquina. Utiliza sistema operativo CP/M, lo que permite un amplio *software*.

**Carcasa.** La carcasa de plástico rígido forma, al cerrarse, un maletín muy fácil de transportar.

***Interface* RS232C.** Permite conectar a la máquina impresoras en serie y cualquier dispositivo que use esta *interface* estándar.

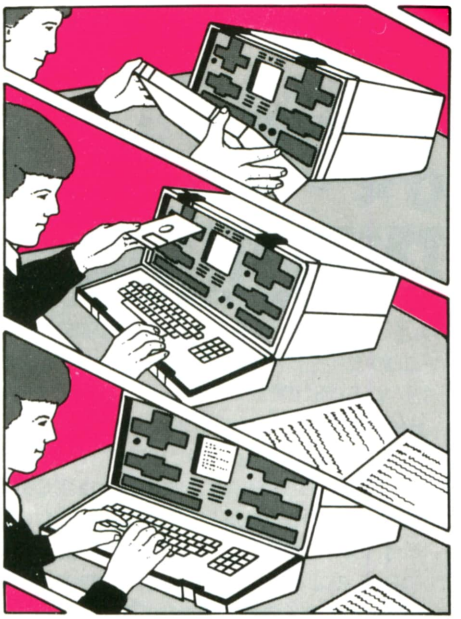
***Interface* IEEE-488.** Permite conectar la máquina al *bus* de instrumentación estándar para intercambio de datos con instrumentos de test.

**Teclado.** El teclado viene inserto en la tapa del ordenador. No obstante, podemos sacarlo y ponerlo donde nos sea más cómodo.. Este teclado, a diferencia de otros portátiles, es del tamaño habitual.

**Pantalla.** La diminuta pantalla (véase Módulo 20) es legible poniéndose bien cerca. Sin embargo, se puede conectar un monitor externo.

**Almacen de discos.** Si se cuenta con unidades de disco, harán falta discos, y en algún sitio hay que llevarlos. Esta máquina tiene espacio para almacenar hasta veinte discos de 5¼ pulgadas (Módulo 22).

***Interface* para monitor.** Permite conectar la máquina a un monitor.



**El portátil tipo maletín.** La máquina aquí presentada se diseñó con la finalidad de que fuese barata y compacta, y pese a ello dispone de una práctica colección de *software* listo para ser usado. Su aspecto exterior es el de un maletín, y al abrirlo (véase la figura de arriba) aparece un teclado completo con una pequeña pantalla (se puede también conectar una pantalla exterior). Cuenta también con dos unidades de disco. La versión básica incluye tratamiento de textos, hoja de cálculo y sistema gestor de bases de datos.

# ¿QUE ES UN SISTEMA INFORMATICO?

Es erróneo pensar que los ordenadores están constituidos por una sola máquina. Son *sistemas* que constan de varios elementos interconectados, teniendo cada cual su misión propia. Conviene tener una idea de cómo interactúan los diversos componentes; primero, los que llevan a cabo el cálculo: la «unidad central de proceso» *(CPU, central processing unit)* y la «memoria» interna del sistema; luego, los elementos que permiten al ordenador comunicarse con nosotros y viceversa, dispositivos de «entrada-salida» *(I/O, input/output)*.

**La parte del sistema que «piensa»**

El elemento más importante de un sistema informático es la unidad central de proceso. La CPU es el «cerebro» del ordenador. Hace el trabajo más arduo y a ella está conectado el resto.

En un macroordenador la CPU es del tamaño de un armario, pero en un micro es un solo componente, el microprocesador o pastilla de silicio. En los micros modernos se usan dos tipos de CPU, de 8 bits y de 16 bits. Los procesadores de 16 bits suelen ser más potentes y soportar más memoria que los de 8.

**Las partes del sistema que «recuerdan»**

La mayor capacidad de memoria de las máquinas de 16 bits hace que su precio sea más elevado,

por lo que muchos ordenadores domésticos llevan procesadores de 8 bits. Pero la CPU por sí sola de nada sirve, pues no hará ningún cálculo a menos que reciba las instrucciones (el programa) e información (los datos) con que debe trabajar. Programa y datos han de estar almacenados en el sistema informático, y a su almacén se le llama memoria (véase Módulo 19).

Hay dos tipos de memoria, RAM y ROM. La RAM se utiliza para almacenar temporalmente programas y datos, es decir, sólo cuando se van a usar y sólo mientras el ordenador está encendido. En la ROM el almacenamiento es permanente, su contenido se fija en el momento de fabricarla y lo normal es que no pueda, por tanto, alterarse.

Los *chips* de ROM y RAM tienen una capacidad de memoria limitada. Para almacenar programas largos y grandes cantidades de datos cuando se apaga el ordenador, se utilizan varios tipos de periféricos de almacenamiento; entre ellos, los magnetófonos, los *floppies,* los discos duros y los cartuchos. Los magnetófonos son, sin embargo, extraordinariamente lentos, y por ello inadecuados para aplicaciones de gestión. De ahí que los *floppies* constituyan el sistema de almacenamiento más generalizado. Hacen todo lo que hace la cinta, pero con mayor rapidez. Los discos duros son aún más eficientes, y de mayor capacidad, pero su precio es todavía elevado, por lo que se emplean casi exclusivamente para aplicaciones de gestión.

### Las partes del sistema que «comunican»

Para comunicarnos con el ordenador tenemos el teclado, el dispositivo de entrada de datos más usual. El teclado de un ordenador es como el de una máquina de escribir, con unas cuantas teclas más. Sin embargo, no hay papel: los caracteres tecleados aparecen en un «monitor», una pantalla parecida a la de un televisor. Con el monitor se ahorra papel, pero no puede disponerse de una

## UN SISTEMA DOMESTICO TIPICO

El mínimo imprescindible para constituir un sistema informático es el ordenador mismo —la CPU y su memoria— más algún recurso para la comunicación entre el ordenador y el usuario —un teclado y una pantalla—, y un medio para almacenar los programas.

La figura nos presenta un sistema informático doméstico típico. La mayor parte de los ordenadores domésticos pueden usar la pantalla de un televisor corriente. Para almacenar programas y datos suele recurrirse al magnetófono. Casi todos los domésticos pueden utilizar prácticamente cualquier tipo de magnetófono, aunque algunos funcionen mejor que otros. Algunos fabricantes, sin embargo, insisten en que se emplee sólo el que ellos proporcionan. Sale más caro, pero suele ser más fiable.

**Pantalla.** La salida del ordenador se presenta en un televisor corriente. La mayor parte de los ordenadores domésticos de hoy en día pueden generar colores. Para usarlos, el televisor ha de ser de color.

**Magnetófono.** Uno de los métodos más populares de almacenar los resultados del ordenador y de introducirle datos

y programas es el *cassette* ordinario (Módulo 21). Aunque sea más lento que los discos (Módulo 22), nos ahorra un aparato especializado.

**Unidad principal.** La CPU (Módulo 17), la RAM (Módulo 19) y el resto de la electrónica están en la unidad principal del ordenador. Aquí se alberga también el teclado (Módulo 20) para comunicarse con él.

# TIPICO PEQUEÑO SISTEMA DE GESTION

**Pantalla.** Todo sistema de gestión precisa una pantalla de buena calidad, de por lo menos 24 líneas de 80 caracteres (véase Módulo 20). Un televisor corriente no es lo bastante bueno.

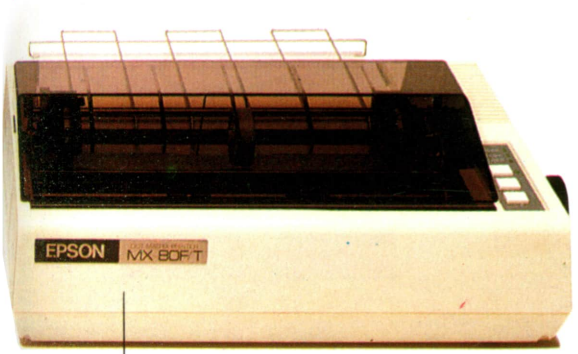
**Unidades de disco.** La rapidez y la comodidad exigen contar con dos unidades gemelas (véase Módulo 22), pues el cassette resulta demasiado lento.

**Unidad principal.** Alberga toda la electrónica y también el teclado, aunque últimamente se tiende a separarlos (véase Módulo 20).

Para usos comerciales se requiere un equipo más sofisticado que el de las páginas 38-39. Las *cassettes* son aceptables para usos domésticos, pero demasiado lentas para aplicaciones de gestión, para lo que se emplean *floppies*. Normalmente se cuenta con dos unidades de disco, pues el realizar copias de seguridad (o *back-up)* de los programas o datos es demasiado farragoso con una sola unidad. Esto es esencial para cuando se deteriore un *floppy*.

En un televisor corriente no pueden aparecer más de 16 líneas de 64 caracteres. En gestión esto resulta insuficiente y es imprescindible contar con un monitor de 24 ó 25 líneas de 80 caracteres. Ello exige utilizar una pantalla profesional, capaz de mostrar una mayor cantidad de texto, con caracteres más nítidos y legibles.

Los usuarios de los sistemas de gestión necesitan imprimir copias en papel. Una impresora matricial suele resultar adecuada cuando la calidad de impresión no es crucial. Para gran calidad habrá que utilizar una impresora de margarita.

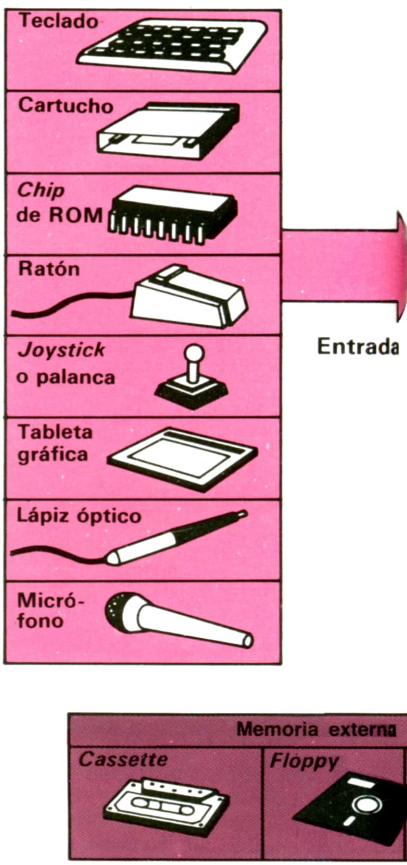



**Impresora.** Toda empresa necesita poner por escrito datos. La mínima solución practicable es una impresora matricial de bajo coste (véase Módulo 23).

salida impresa permanente. Para ello se necesita una «impresora» (véase Módulo 23).

Las impresoras modernas se clasifican en dos grandes grupos, las matriciales y las de margarita. Las matriciales componen los caracteres sobre el papel a partir de un conjunto de puntos diminutos. Su precio es bajo y su velocidad grande, pero la calidad de los caracteres impresos es inferior a

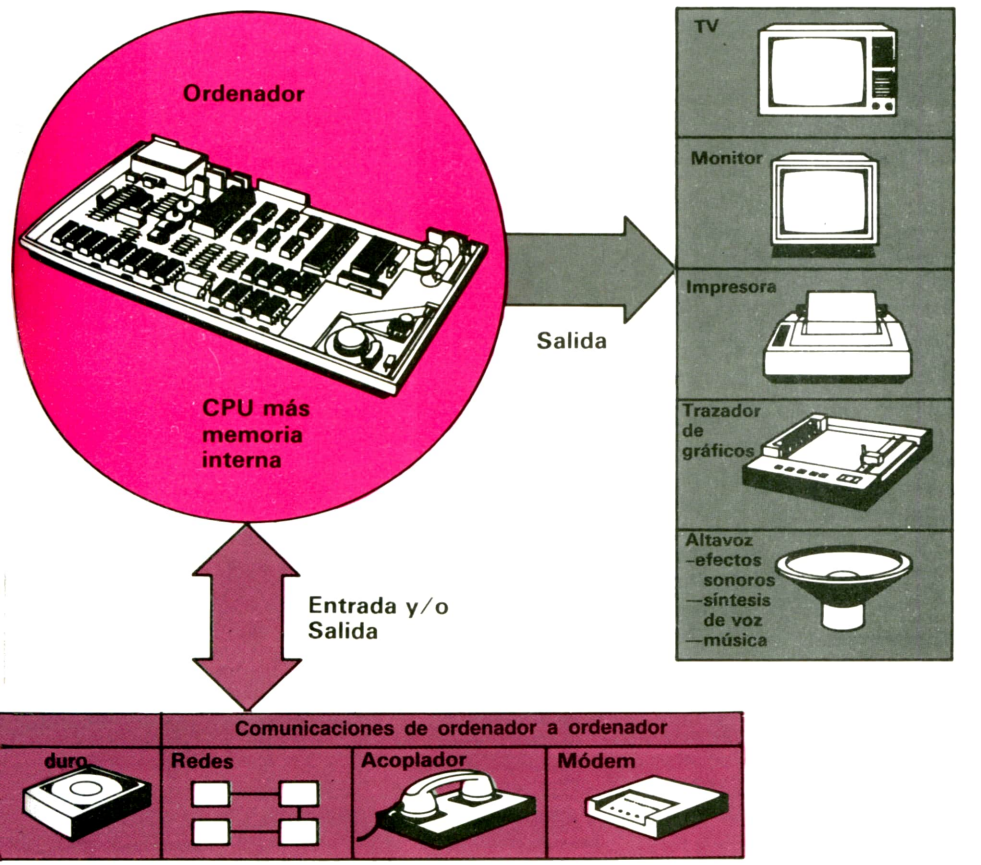
## EL ARBOL GENEALOGICO




La CPU y su memoria pueden conectarse a una amplia gama de dispositivos. Algunos son esenciales, y figuran en todos los ordenadores: teclado, pantalla y sistema de almacenaje. Otros son para usos especializados, como los periféricos gráficos, y los cartuchos de *software*. También los ordenadores se conectan entre sí, vía redes locales o por teléfono, usando módems y acopladores.

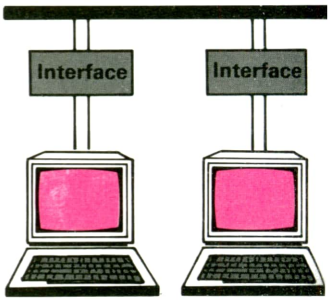
la que consiguen las impresoras de margarita, que, no obstante, son más caras y lentas.

Existen también otros dispositivos de entrada-salida más exóticos (véase Módulo 24). Uno muy frecuente es el *joystick* o palanca de rótula, que se utiliza sobre todo en juegos. También hay dispositivos más serios, para comunicaciones o gráficos.

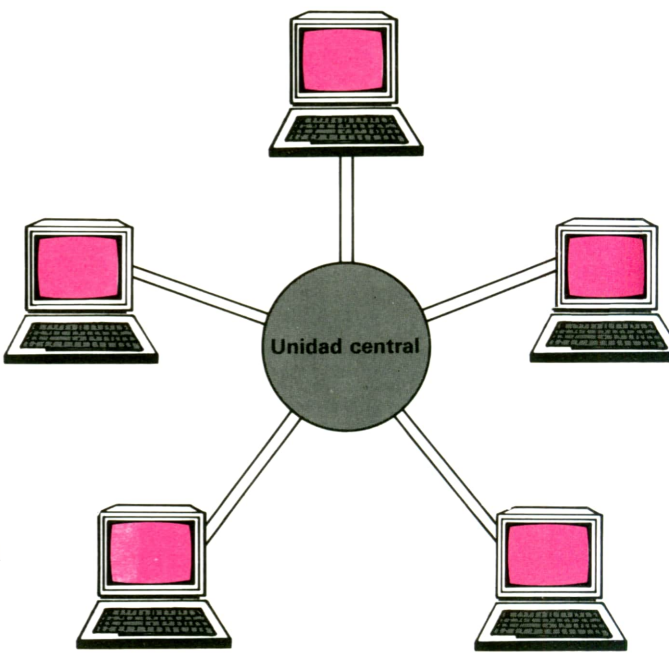


## REDES

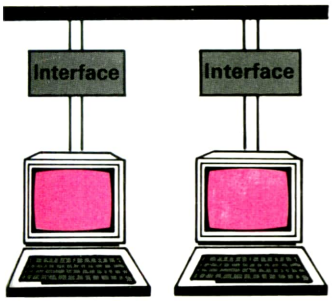
En la época de los ordenadores grandes se consideraba antieconómico que los utilizase una sola persona a un tiempo. Se descubrió que con un *software* podían trabajar para diferentes usuarios.
En los micros, la CPU no suele tener potencia suficiente para llevar a cabo este trabajo multiusuario. Como la CPU es tan barata, cada usuario puede tener su propia unidad. Los elementos más caros, como discos duros o impresoras, estarán más racionalmente utilizados, por contra, si acceden a ellos varias personas. Y esto es posible al conectar varios micros formando una «red». La idea es sencilla: un grupo de máquinas interconectadas por cable. Cuando un ordenador se quiere comunicar con otro envía una señal para identificar al destinatario. Este estaba «a la escucha», y al oír su señal contestará.



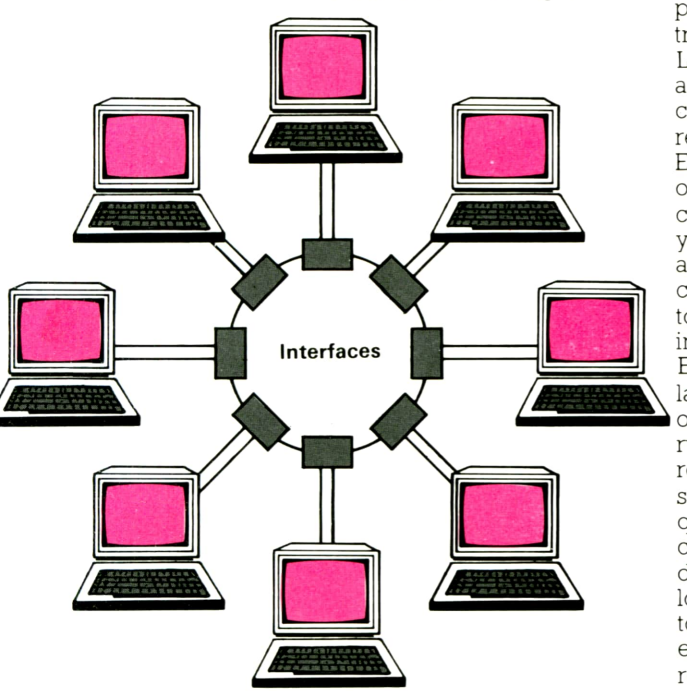

**La red en *bus*.** Cada dispositivo escucha las señales que circulan por el *bus*. Cuando un dispositivo reconoce un mensaje se lo envía a su ordenador. En esta red puede retirarse cualquier elemento sin que se vea afectado el funcionamiento de los demás.




**La red en estrella.** En esta red toda la información ha de pasar por un controlador central del sistema. El controlador examina todos los mensajes, averigua quién es el destinatario de cada uno, y lo envía en la dirección adecuada. Sin embargo, todo el sistema se viene abajo si el controlador central falla.




**La red en anillo.** Este sistema lleva incorporado determinado número de *interfaces*, cada una de las cuales intercepta y examina los mensajes para ver si el destinatario es su ordenador. Si no lo es, se pasa el mensaje a la siguiente *interface*, donde volverá a ser examinado antes de seguir.




En la práctica, la tarea resulta más complicada, y se han ideado métodos diversos: las conexiones en *bus*, en estrella y en anillo. En la conexión en *bus* todos los dispositivos están unidos al mismo cable a través de una *interface*. Estas *interfaces* están permanentemente a la escucha de las señales que circulan por el canal común *(bus)*, y cuando una de ellas recibe su señal, transmite la información a su ordenador. Cuando un ordenador desea transmitir información, su *interface* comprueba si el canal común está libre y, si es así, permite a su ordenador transmitir.

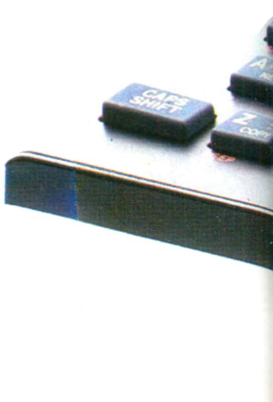
La red en estrella se apoya en un dispositivo central que gobierna toda la red: el controlador de red. Este decide qué ordenador debe recibir una comunicación determinada y la encauza en la dirección apropiada. Pero si el controlador se estropea, toda la red queda inutilizada.

En la conexión en anillo, la *interface* de cada ordenador se sitúa en la ruta de los datos que recorren el anillo. Los datos se envían en «paquetes» que comienzan por un código de identificación del destinatario. Cada uno de los ordenadores intercepta todos los paquetes, y si no es el destinatario, lo retransmite al siguiente.

# UN ORDENADOR POR DENTRO

Levantando la tapa podremos ver el interior de un microordenador y enterarnos de lo que contiene. El modelo de la derecha —llamado Sinclair ZX Spectrum— es un típico representante de la nueva generación de ordenadores domésticos, pequeños y baratos. Ofrece colores, gráficos, sonido y una potencia de cálculo que hubiera parecido impensable hace sólo unos pocos años en una máquina de tan reducido tamaño y bajo precio.

Separado el teclado del resto de la carcasa observamos (páginas 48-49) que los componentes están montados en una única tarjeta de circuito impreso. El efecto es de una gran sencillez; se diría que el ordenador no consta más que de un puñado de *chips,* un modulador de UHF (para enviar las imágenes a la pantalla de televisión o al monitor), un diminuto altavoz y una serie de condensadores, resistencias y diodos situados entre el entramado de pistas del circuito impreso.




Los *chips* son los componentes clave para el funcionamiento del ordenador. Son bloques rectangulares de plástico negro conectados a la tarjeta por las patas metálicas alineadas en sus bordes. El más importante de estos *chips* es el llamado Unidad Central de Proceso o CPU. Sin embargo, la CPU por sí sola no sirve de nada. Precisa de la «memoria» porque, como tiene que haber unas instrucciones que le digan lo que tiene que

hacer, en algún sitio tiene que almacenarlas, lo mismo que la información o los datos sobre los que va a trabajar. También le es necesario almacenar los resultados obtenidos antes de transmitirlos al mundo exterior. La memoria es, por tanto, el segundo componente en importancia.

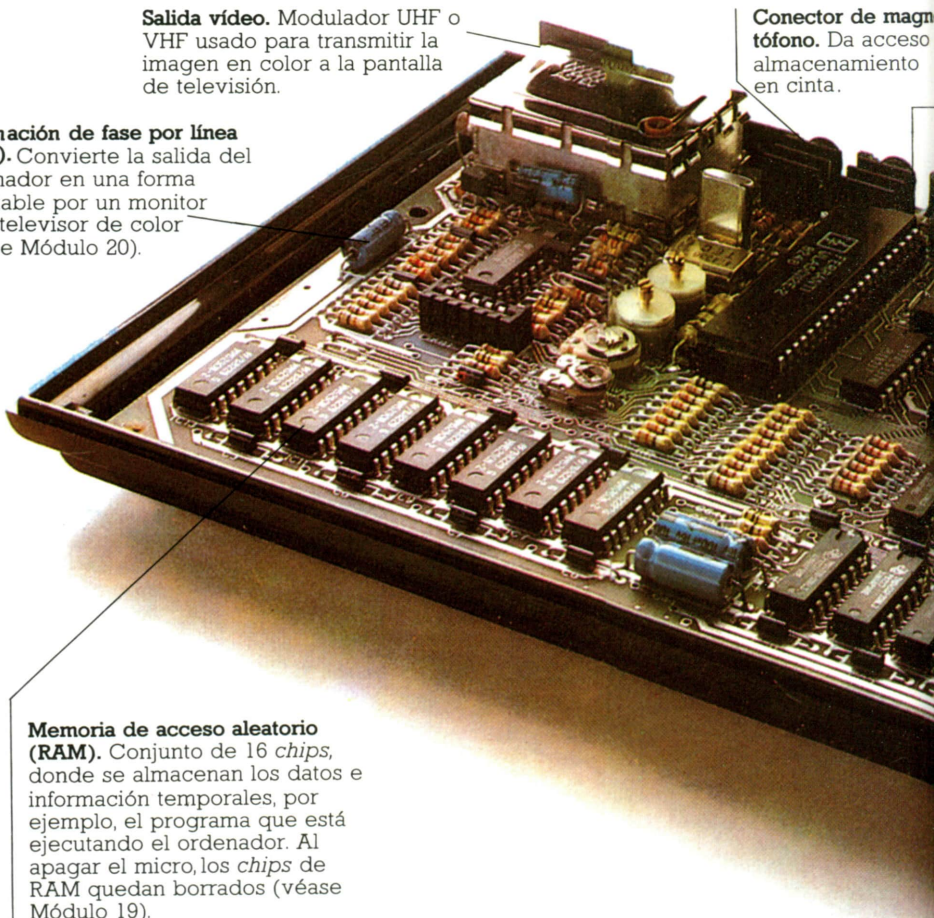
### Los *chips* de memoria

Ya veremos, con precisión, cómo funciona la memoria, y los diferentes tipos que existen. Por el mo-

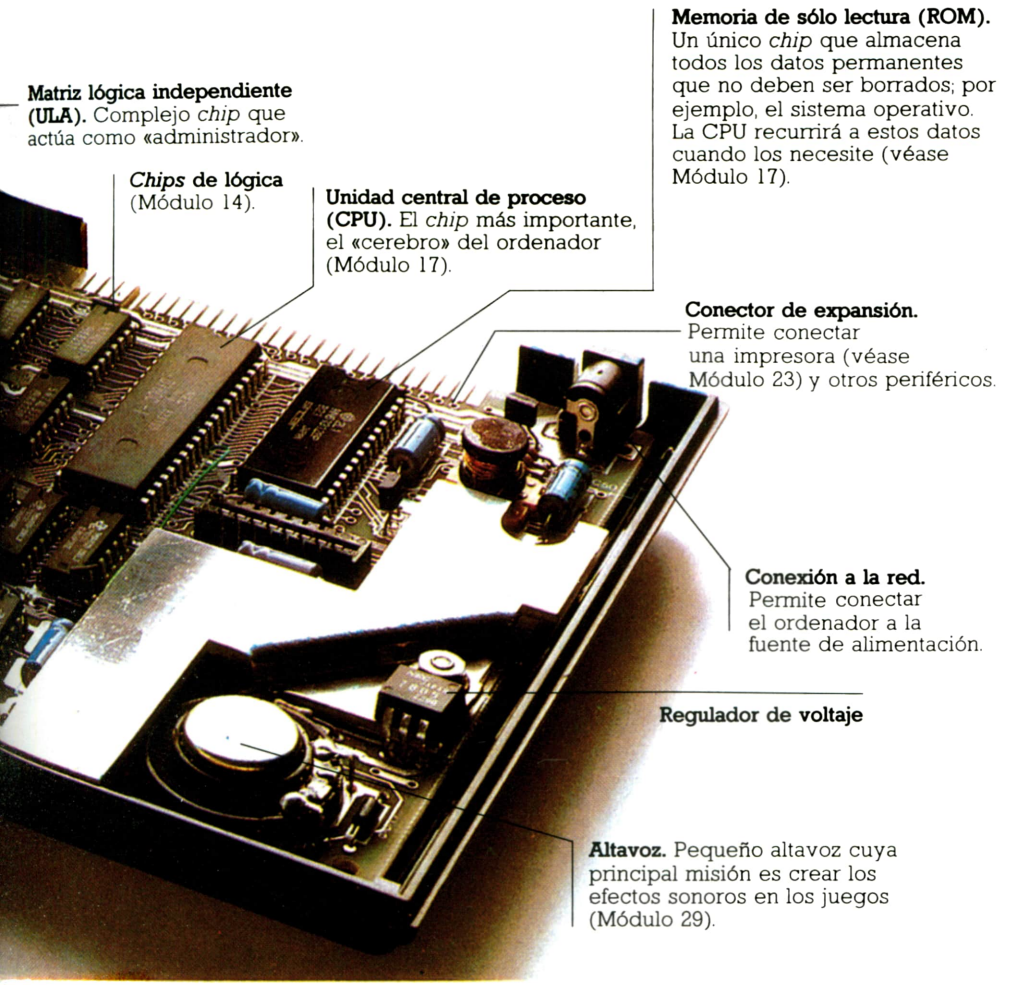
**EL ZX SPECTRUM DE SINCLAIR**

Lanzado inicialmente en Inglaterra en 1982, este micro no es mucho mayor que un libro de bolsillo. Todos sus componentes electrónicos van montados en un único circuito impreso, visible al levantar el teclado. Existen dos versiones: una con 16 kilooctetos de RAM y otra (la que aquí se muestra) con 48. Ambas llevan la misma versión especial de Sinclair del lenguaje Basic en el *chip* de ROM. Muchas de las instrucciones que en Basic son palabras pueden obtenerse pulsando una sola tecla. El ZX Spectrum puede conectarse a un magnetófono.

**Teclado.** Unido normalmente a la tarjeta principal por dos cables planos, el teclado es el habitual de las máquinas de escribir más algunas teclas especiales (véase Módulo 20).

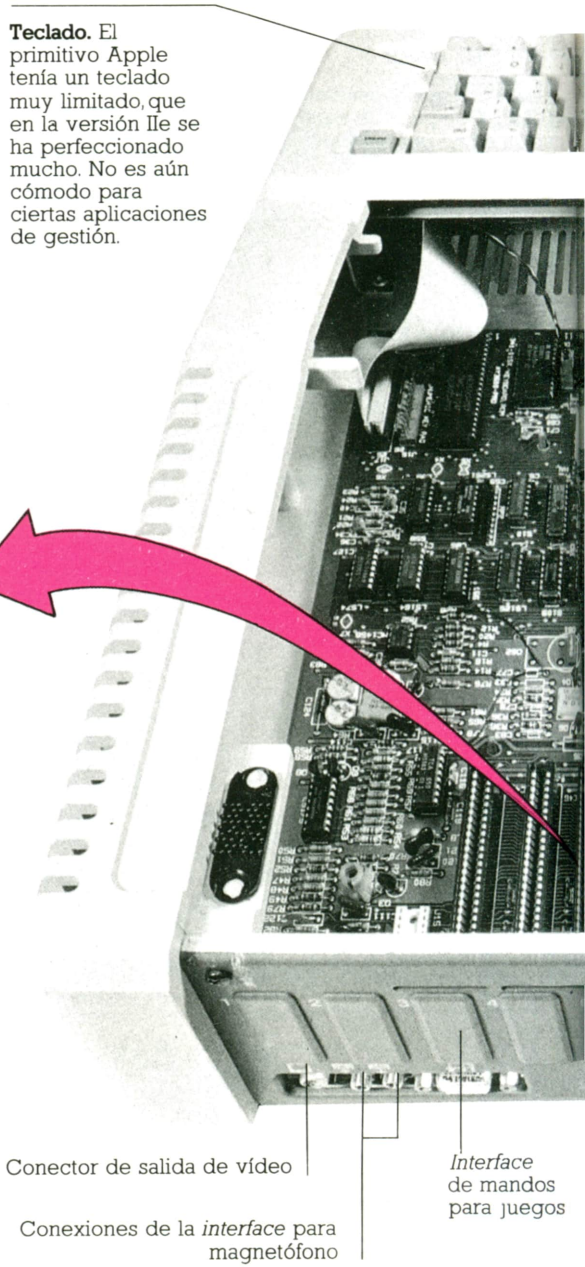



mento baste decir que hay dos tipos básicos: en primer lugar, la que se usa para almacenar instrucciones e información cuando el ordenador está apagado (Memoria de Sólo Lectura, o ROM, *Read Only Memory)* y, en segundo, la memoria más rápida donde se mantienen el programa y los datos sobre los que el ordenador está trabajando en un momento dado (Memoria de Acceso Alea-




torio o RAM, *Random Access Memory).* En el Spectrum, toda la ROM está contenida en un único *chip* grande. Ahí residen muchas de las instrucciones que precisa la CPU para trabajar. En este caso figuran entre ellas el sistema operativo de la máquina y el lenguaje Basic. De esta manera, el *chip* de ROM es capaz de «interpretar» un programa escrito en Basic, traduciéndolo a una

# APPLE IIe



**Teclado.** El primitivo Apple tenía un teclado muy limitado, que en la versión IIe se ha perfeccionado mucho. No es aún cómodo para ciertas aplicaciones de gestión.

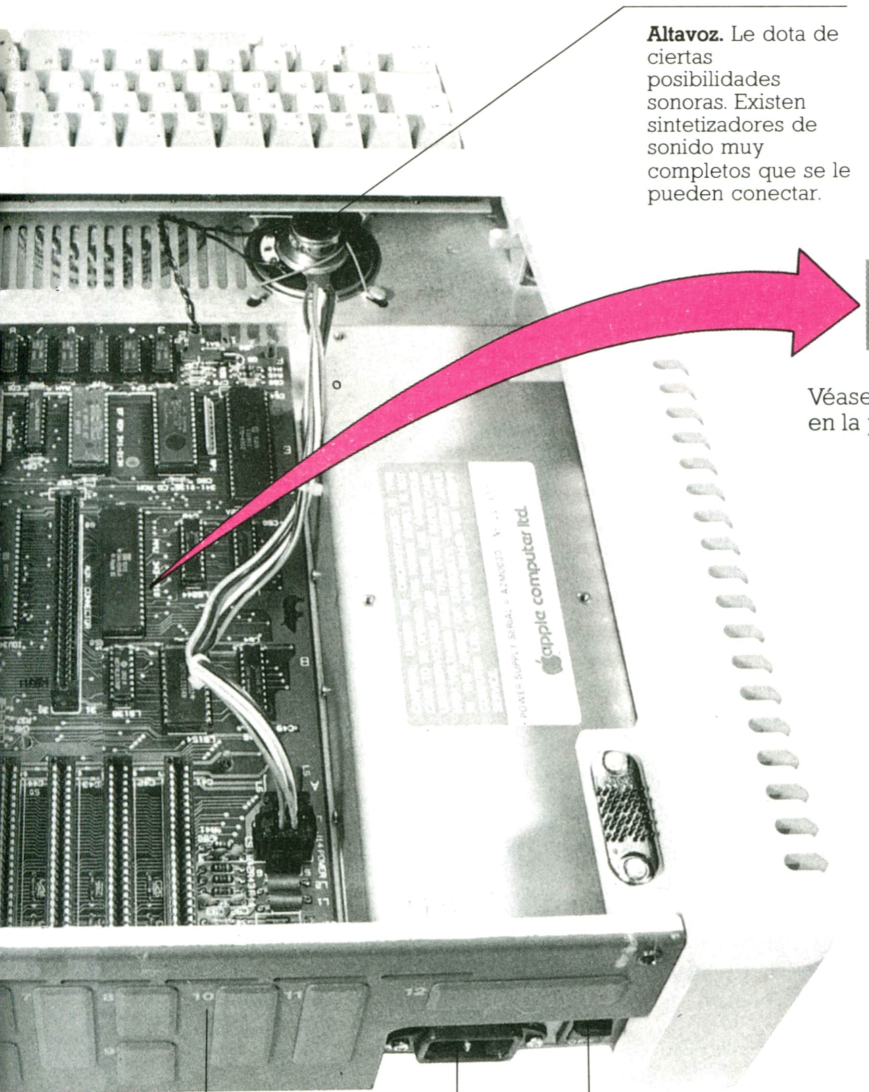
Véase el detalle en la página 52.

El Apple II es uno de los micros pioneros y de mayor éxito. El IIe cuenta con nuevos accesorios, mejoras en el teclado y una construcción interna más sencilla. Como el Apple II lleva tanto tiempo en el mercado y se han vendido cientos de miles en estos años, dispone de una amplia variedad de *software* y *hardware* directamente incorporable. Casi todo lo que se puede hacer con un microordenador lo podemos hacer con alguna de las máquinas de la familia Apple, aunque no necesariamente de forma más sencilla o económica que con otros micros de categoría semejante.

Conector de salida de vídeo

Conexiones de la *interface* para magnetófono

*Interface* de mandos para juegos



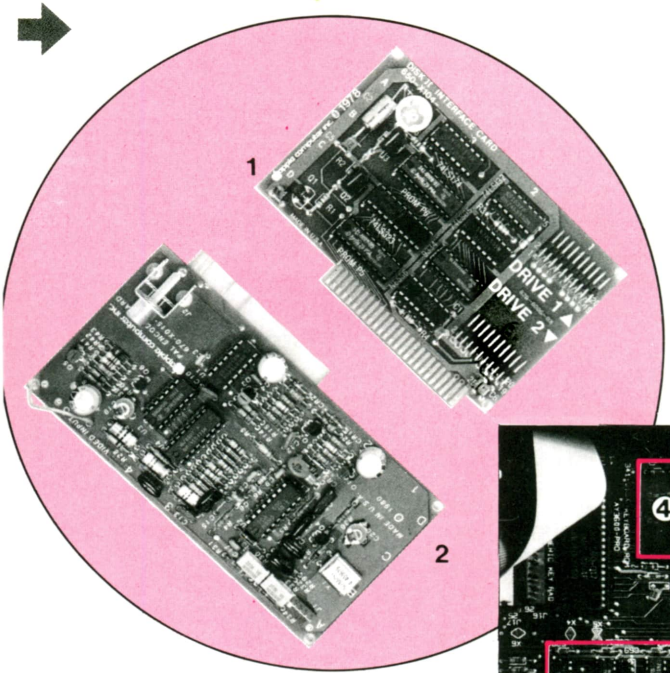

**Altavoz.** Le dota de ciertas posibilidades sonoras. Existen sintetizadores de sonido muy completos que se le pueden conectar.

Véase el detalle en la página 52.

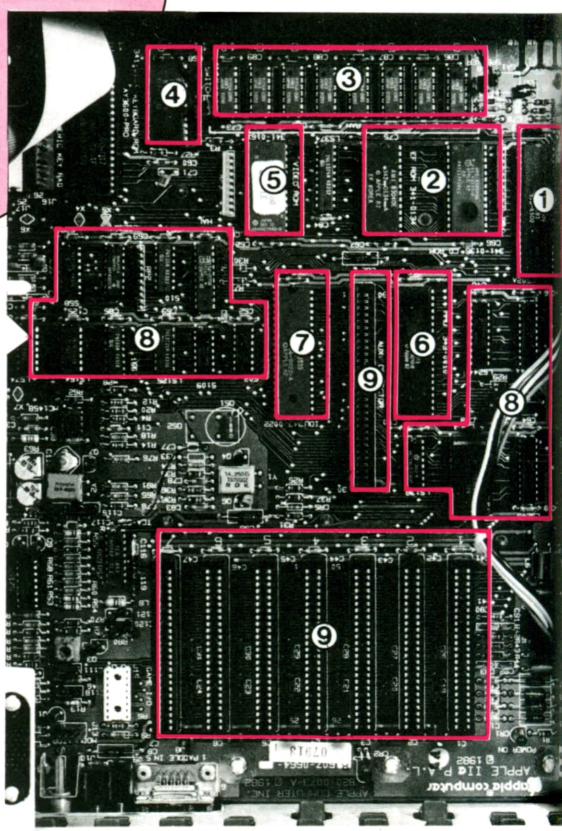
Interruptor de encendido y apagado

Paneles desmontables para más conectores de entrada-salida

Conector de la fuente de alimentación




**Tarjetas de circuito impreso.** El Apple cuenta con varios conectores internos donde pueden colocarse tarjetas suplementarias, que nos darán acceso a una amplia gama de complemento, como unidades de disco, etc. Es muy extensa la colección de tarjetas disponible, y las dos que aparecen en la figura son un controlador de disco (1) y una tarjeta de vídeo color (2).




**El Apple por dentro.** Los componentes principales son fáciles de distinguir. Toda la máquina está construida en una única tarjeta de circuito impreso. El «corazón» del Apple es el *chip* de la CPU (1), un microprocesador 6502. El *software* de control reside en las ROM (2) y existe también una zona de RAM (3). El *software* para generar caracteres en la pantalla y para manejar el teclado reside en ROM especiales (4 y 5). Dos grandes *chips* (6 y 7) controlan la memoria y la entrada-salida. Distintos grupos de puertas lógicas (8) realizan funciones variadas en el sistema, y una serie de conectores hembras (9) permiten enchufar tarjetas con controladores de disco, *interfaces* de entrada-salida suplementarias o más RAM.

forma que la CPU pueda «comprender» y ejecutar.

La RAM del ordenador la componen una serie de *chips* pequeños. Este modelo, que lleva 48 kilooctetos de RAM, cuenta con 16 *chips* independientes. Ahí reside la información que el ordenador necesita sólo temporalmente, por ejemplo los programas que se están ejecutando. Cuando se apaga la máquina, todos los *chips* de RAM quedan borrados. El ordenador «olvida» su contenido.

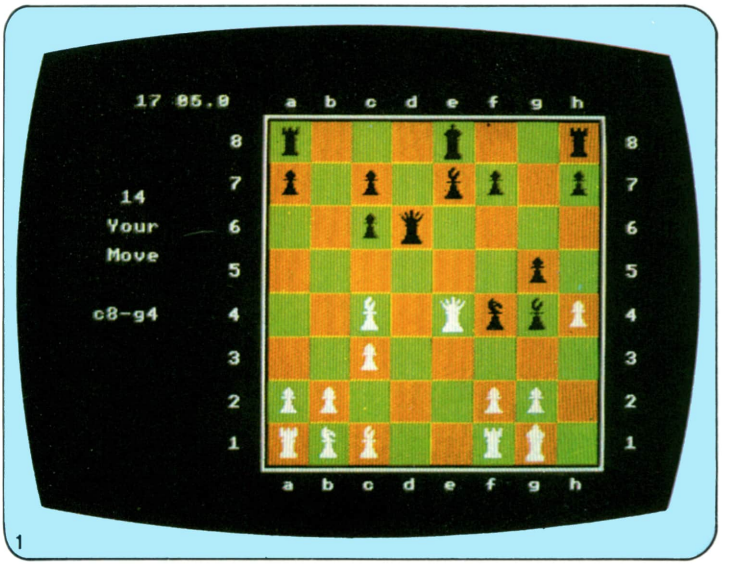
Puede que echar un vistazo al interior de nuestro ordenador resulte interesante, pero en la mayor parte de las máquinas el hacerlo invalidaría la garantía ofrecida por el fabricante. Además, algunas máquinas no resultan tan fáciles de abrir y podrían romperse (lo que es especialmente aplicable al Spectrum).

**¿Son iguales todos los ordenadores por dentro?** Algunos microordenadores como el Apple II de las páginas 50-51, el PC de IBM y el QX-10 de Epson, están diseñados de modo que resulta fácil el acceso a su interior. Todas estas máquinas —y otras— llevan dentro una serie de conectores en los que enchufar diversas tarjetas de circuito impreso para potenciar la versatilidad de la máquina —por ejemplo, más memoria (véase Módulo 19), más canales de entrada-salida (véase Módulo 20), o colores para juegos (véase Módulo 29). Estos conectores, que aparecen en la parte trasera del Apple de la ilustración, lo convierten en una máquina ideal para una gran variedad de aplicaciones de gestión, aparte de las domésticas.

# APLICACIONES DE LA INFORMATICA

Se equivoca quien piense que los ordenadores lo saben hacer todo. No es ya que sean incapaces de hacer nada si no se les han dado las instrucciones adecuadas; es que sus poderes son limitadísimos. El ordenador es una máquina sumamente versátil, pero no puede actuar en el mundo físico a menos que se le dote de conexiones especiales. Sin embargo, le resultará extremadamente sencillo resolver un complejo rompecabezas matemático, decir la hora, jugar al ajedrez, o ayudarnos a escribir una carta. Este es el secreto del éxito del ordenador: se trata de una máquina «en blanco» que, dentro de ciertos límites, puede llevar a cabo una amplia gama de tareas.

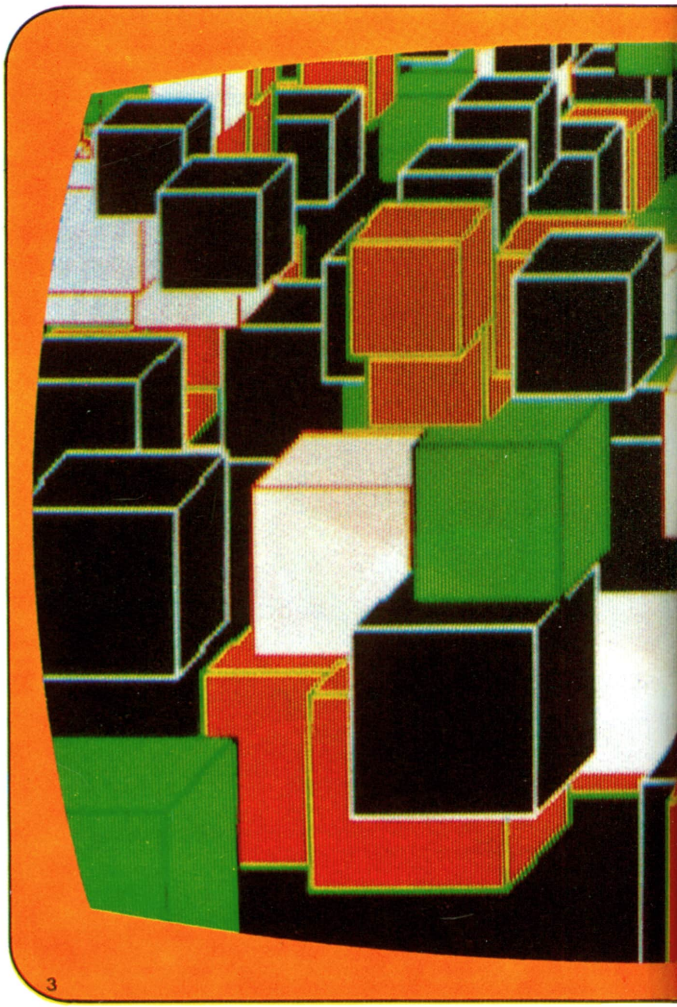
Una de las principales bazas del ordenador es la velocidad a que trabaja. Pidámosle que sume una columna de mil números y lo hará en un tiempo increíblemente menor que una persona. Análogamente, presentémosle un libro entero en forma «legible» para él —por ejemplo en disco (véase Módulo 22)— y pidámosle que busque determinada frase. Su aureola de poder deriva de la velocidad tremenda con que es capaz de llevar a cabo instrucciones muy sencillas, lo que le permite aparentar unas posibilidades muy por encima de las nuestras. Todo lo que hace un ordenador lo podríamos hacer nosotros también, aunque tardando más y tal vez con menor precisión.




1

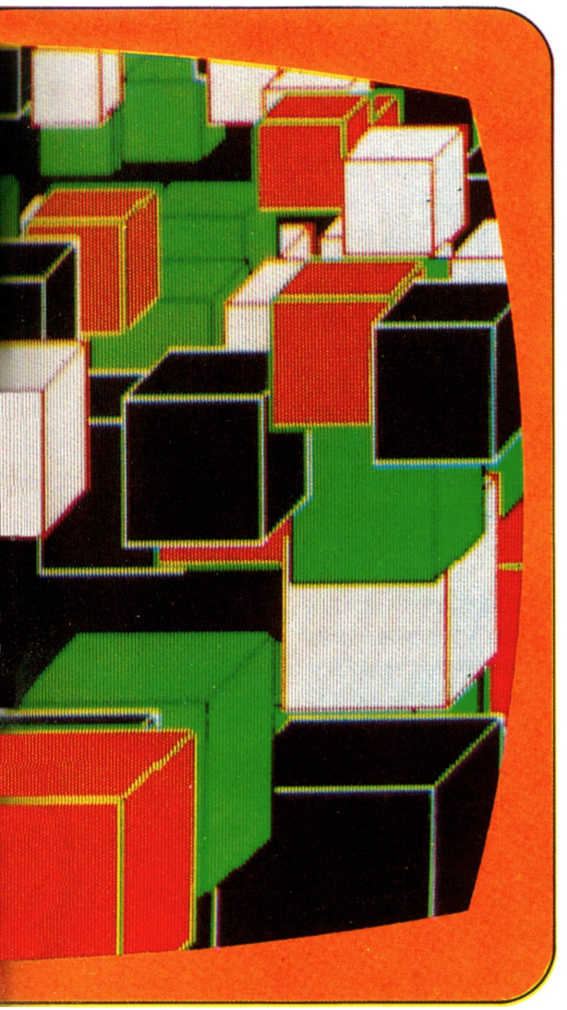
2

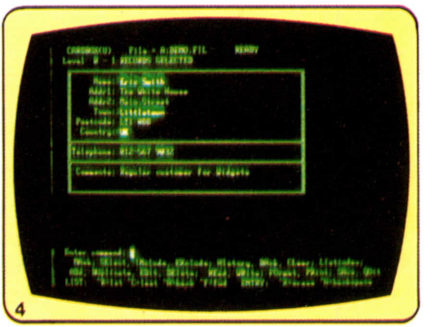
Los juegos (1) son uno de los usos más extendidos. El Videotex (2), un método de obtener información a través del teléfono.



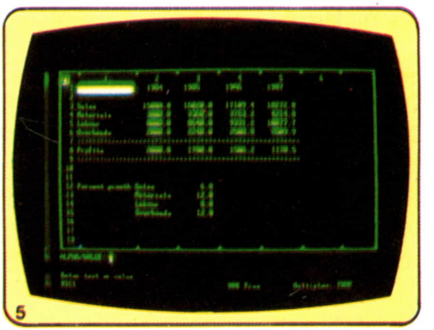
Las gráficas (3) tienen su papel en las aplicaciones de empresas. Los sistemas de gestión de bases de datos (4) organizan la información de forma estructurada, para que el micro la pueda manejar a nuestro antojo. La «hoja de cálculo» (5) permite proporcionar al ordenador una serie de números y las relaciones que guardan entre sí. Luego, él calculará instantáneamente los efectos que produce en los demás la alteración de uno de ellos. Con el procesamiento de textos (6), los días de la máquina de escribir están contados.

Sin embargo, la situación inversa no se cumple, es decir, el ordenador no sabe hacer todo lo que nosotros hacemos. Es incapaz de «razonar», de hacer conjeturas o de manifestar siquiera una de las cualidades emotivas humanas. Ni tiene conciencia de sí mismo, ni personalidad, pese a las con-

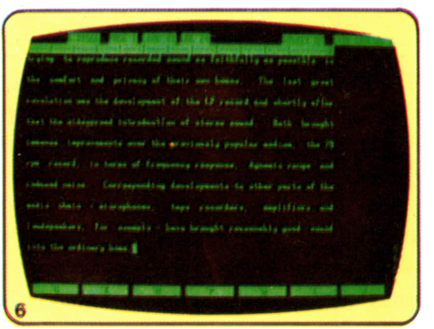




4



5



6

notaciones del término «inteligencia artificial» (véase Módulo 37).

Las posibilidades del ordenador dependen mucho de la habilidad con que se haya escrito su *software* (véase Módulos 25-36), y esa tarea corresponde en lo fundamental a los seres humanos.

# HISTORIA DE LA INFORMATICA

En sólo cuarenta años se han sucedido cuatro generaciones de ordenadores. En este breve período de tiempo hemos sido testigos de la transformación de los lentos y voluminosos ordenadores de la Segunda Guerra Mundial en las máquinas compactas y ultrarrápidas de nuestros días. Y entre bastidores aguarda ya la quinta generación, una nueva raza de ordenadores más potentes aún.

Es tan acelerado el paso con que los ordenadores nos acercan al futuro que quizá resulte difícil creer que no tengan más de cuarenta años. La tecnología del *microchip,* responsable de la velocidad y extremada pequeñez de las máquinas modernas, tanto como de su fiabilidad y bajo precio, tiene poco más de diez años. Pero la fecha de nacimiento que con más propiedad puede aplicarse a los ordenadores es la década de los 40, pues fue entonces cuando aparecieron en la forma en que hoy los conocemos. Aquellos ordenadores eran máquinas digitales electrónicas. Si los ordenadores pueden trabajar tan deprisa y acometer tareas de tan enorme complejidad, es porque tratan datos en forma de dígitos o números y porque están accionados por circuitos electrónicos.

Las exigencias de la guerra trajeron al mundo al ordenador de manera un tanto precipitada, aunque bien formado, porque su período de gestación había sido muy largo. Los orígenes de los or-

denadores, en tanto que máquinas de manipular números, se remontan a siglos atrás. También pueden trabajar con palabras y realizar operaciones lógicas de muchos tipos, pero dentro del ordenador todo se traduce a cadenas de dígitos.

**Primitivas herramientas de cálculo**

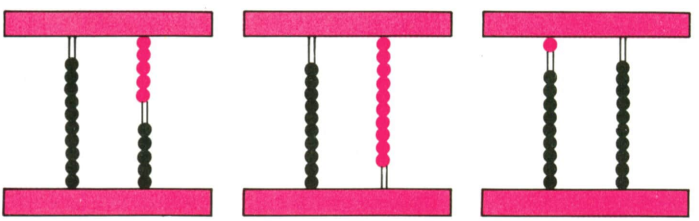
Aparte de la simplísima aritmética mental, nuestras primeras herramientas de cálculo debieron ser nuestros propios dedos. Pudiera pensarse que el contar con los diez dedos llevaría inexorablemente a un sistema de numeración decimal, en el que los números se representan mediante grupos de diez símbolos colocados por orden de valor. Nosotros utilizamos de este modo los números del 0 al 9 con inmensa facilidad, pero a los antiguos las ventajas de tal sistema no les resultaron tan obvias. Por el contrario, idearon sistemas tan difíciles de manejar como los números romanos, en el que se eligieron arbitrariamente ciertas letras para representar números. Imaginémonos tratando de dividir DCCL entre LXXV.

Por suerte, las civilizaciones antiguas contaban con un dispositivo de cálculo ideado en Babilonia hace unos 5.000 años, el ábaco. El ábaco, al igual que el ordenador, manipula números en forma de códigos formados por la presencia o ausencia de objetos materiales, las cuentas ensartadas en unos alambres o varillas sujetos a un bastidor. En un ordenador se trata de pulsos eléctricos o regiones de magnetismo. Las configuraciones de cuentas de un ábaco representan un número en un sistema numérico en el que el valor depende de la posición: la cuenta valdrá uno, diez, cien, etc., según la posición que ocupe en el bastidor la varilla en que está inserta. A medida que se desplazan las cuentas para hacer un cálculo, se obtiene un cero corriendo todas las cuentas a un extremo de la varilla y llevándonos una a la varilla siguiente. En los ordenadores, los pulsos electrónicos salen y entran de los circuitos para representar el cero y llevarse una de una manera análoga.

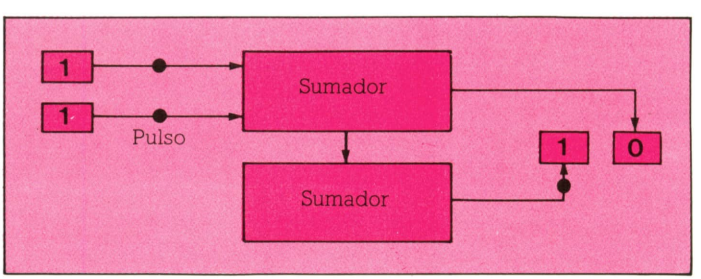
Aunque las civilizaciones egipcia, griega y romana emplearon el ábaco, no contaron con un sistema numérico en el que el cero tuviera un símbolo. Este recurso esencial del cálculo parece haber tenido su origen en la India o haber llegado allí desde Babilonia, y acabó por ser redescubierto por los árabes en el siglo IX. Cuatro siglos después llegaba el cero a Europa.

## EL ABACO Y EL ORDENADOR

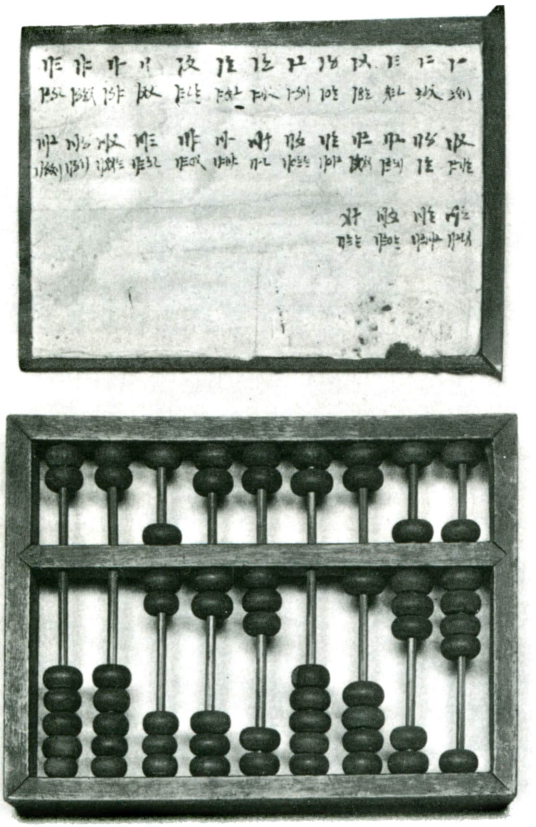
**Sumar con un ábaco.** En el alambre de la derecha (columna de las unidades) se suben dos grupos de 5 cuentas para indicar 5+5 Como con esto se completa el alambre, se vuelven a bajar las 10 cuentas para tener 0 y se sube una del siguiente alambre obteniéndose el «10».



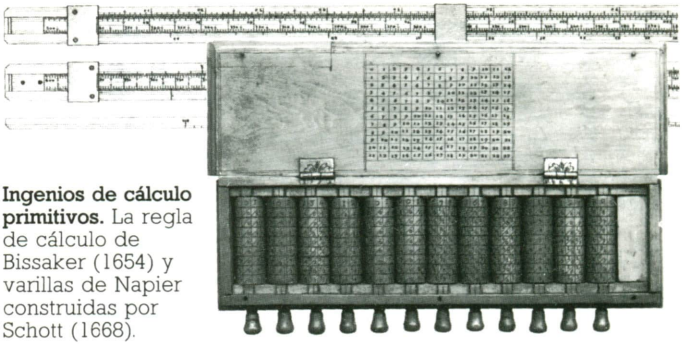
**Sumar con un ordenador.** En vez de cuentas y alambres,en el ordenador se usan pulsos eléctricos y circuitos de adición llamados «sumadores» (Módulo 15). En el primer sumador se suman 1 y 1 (binario). El sumador recibe dos pulsos, y envía un 0 (ausencia de pulso) al resultado y 1 (pulso) al siguiente sumador. Este sumador no recibe más pulsos, y por tanto envía un pulso al resultado. El resultado completo es 10. Convertido de binario a decimal: 1+1=2.



**El ábaco.** La más antigua de las máquinas de calcular, del 3000 a. de C. El número que muestra el ábaco de la figura es el 7.230.189.



Aunque se contaba ya con un sistema numérico adecuado, con el que se podía trabajar en problemas complejos, a muchos les resultaba difícil de dominar, sobre todo en multiplicaciones y divisiones. En su ayuda vino el matemático John Napier, que en 1614 inventó los logaritmos. Para multiplicar números, basta buscar en las tablas sus logaritmos, sumarlos, y buscar luego cuál es el número cuyo logaritmo vale esa suma. La división se efectúa restando los logaritmos. Un método más sencillo de usar estos logaritmos llegó en 1621, año en que el matemático inglés William Oughtred inventó la regla de cálculo.



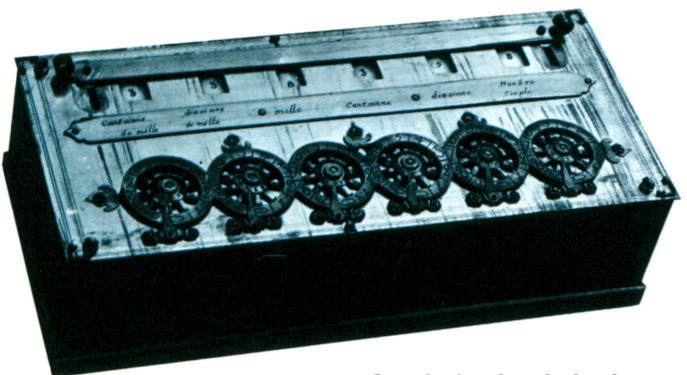
**Ingenios de cálculo primitivos.** La regla de cálculo de Bissaker (1654) y varillas de Napier construidas por Schott (1668).

## Las primeras máquinas de calcular

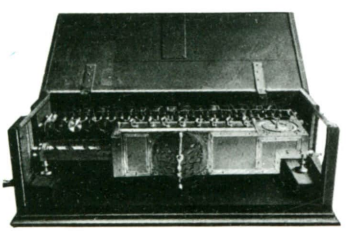
El siguiente paso importante hacia el ordenador debía ser una máquina de calcular verdadera, máquina que no tardó en llegar. La primera la construyó Wilhelm Schickard en Tubinga (Alemania) hacia 1623. Era análoga a la primera calculadora verdaderamente operativa, inventada por el científico francés Blaise Pascal en 1652, sin saber del trabajo de Schickard. Pascal, persuadido de que los cálculos rutinarios no eran dignos de personas, se embarcó en la construcción de una máquina capaz de realizarlos. Fabricó más de cincuenta modelos antes de darse por satisfecho, pero por desgracia el éxito financiero no le acompañó. A los empresarios no les tentaba la idea de cambiar un personal infrapagado por una costosa máquina, y tampoco los empleados de oficina simpatizaban con algo que representaba una amenaza para su medio de vida.

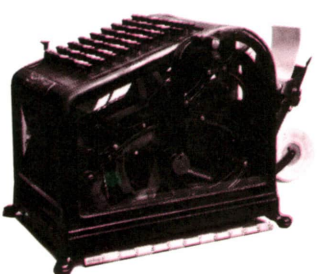
A partir de entonces la historia del ordenador está jalonada por otro genio que, como Pascal, hizo grandes progresos en diversos campos. Fue el filósofo y científico alemán Gottfried Leibniz, quien supuso que una máquina de calcular de múltiples usos capaz de hacer multiplicaciones y divisiones atraería tanto a los matemáticos como a los contables. Leibniz estudió la calculadora de Pascal y las construidas por el inventor inglés sir Samuel Morland posteriormente, siguiendo princi-

pios análogos. Después ideó un brillante método de lograr que tales calculadoras efectuasen reiteradas restas y sumas, con lo que serían capaces de multiplicar y dividir. Para ello Leibniz inventó el contador de pasos, una larga rueda dentada cilíndrica con nueve dientes de diferentes longitudes. Accionaba una rueda corriente más pequeña que, a su vez, operaba el resto de la maquinaria para llevar a cabo los cálculos. La rueda pequeña se colocaba mediante la manilla en un dial,que indicaba por qué número se quería multiplicar o dividir. Adoptaba entonces una posición tal en la rueda multiplicadora que sólo la accionaban el mismo número de dientes de esta rueda. Leibniz inventó su calculadora en 1694, pero no terminó de funcionar bien debido a la falta de precisión de las piezas mecánicas.



**La máquina de calcular de Pascal.** Fiel reproducción de la máquina de Pascal de 1652. El original se halla en el Conservatoire des Arts et Métiers de París.

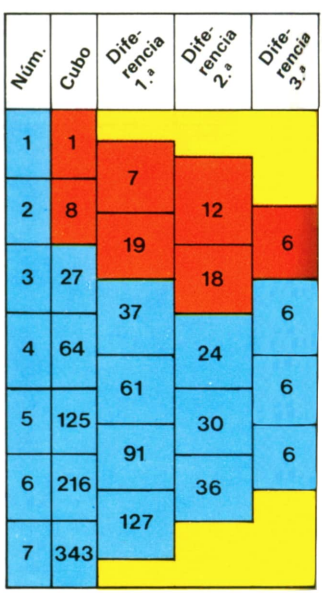


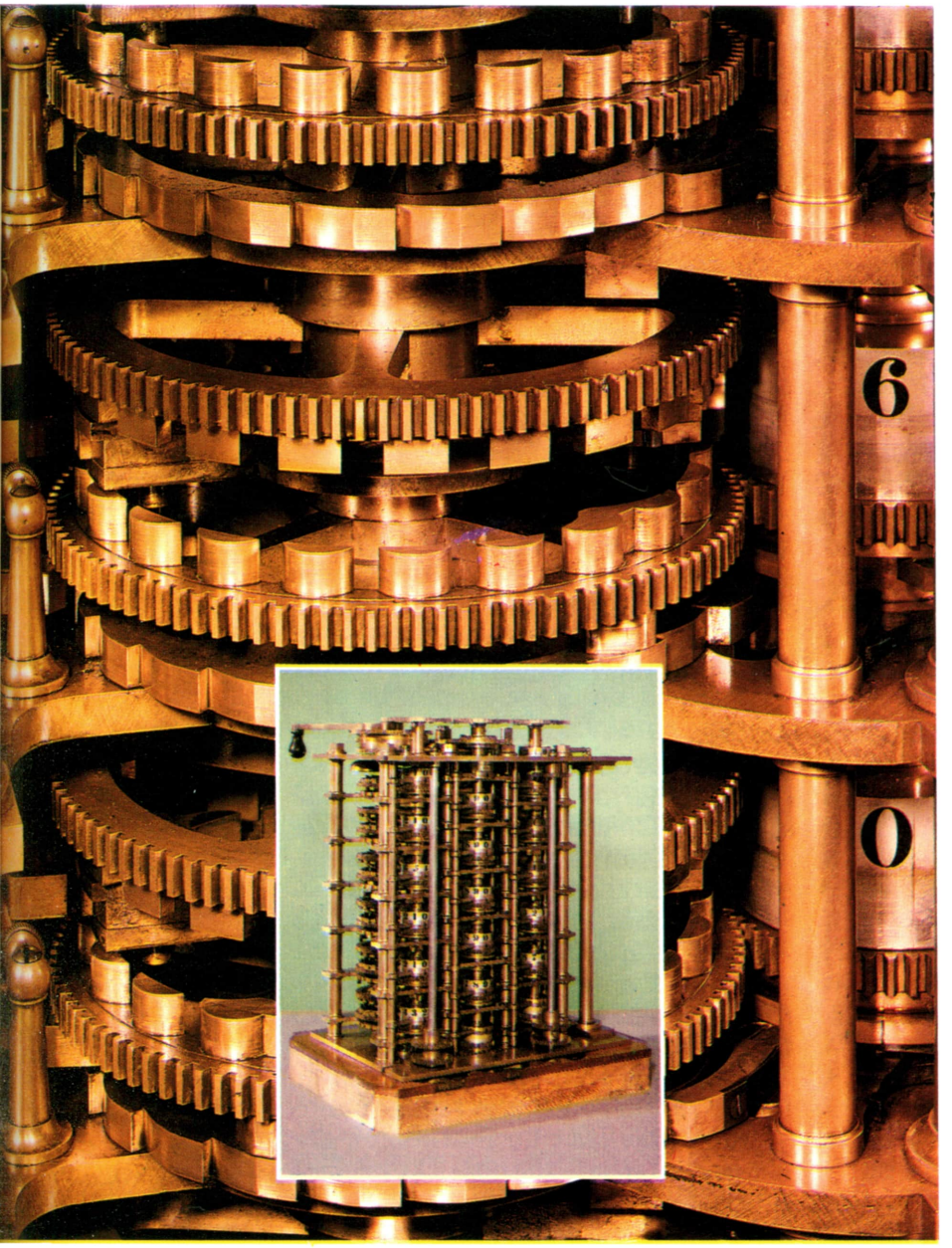


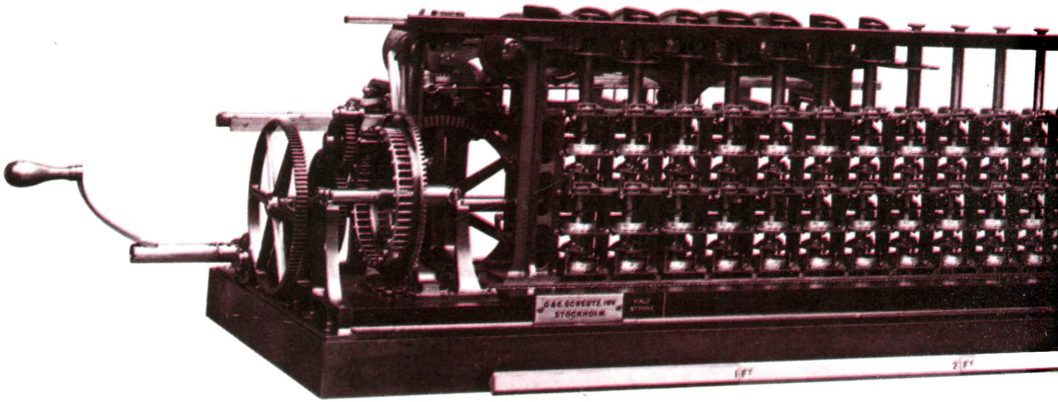
**La máquina de calcular de Leibniz y la máquina de sumar e imprimir de Burroughs.** La máquina de Leibniz (arriba) de 1694. Máquina de Burroughs (derecha), 1913.

# EL METODO DE LAS DIFERENCIAS

Las tablas de logaritmos que se proponía compilar Babbage con su Máquina de Diferencias (derecha) vienen dadas por complejas ecuaciones algebraicas. El método de las diferencias simplifica los cálculos. Tomemos una tabla de los cubos de los enteros. Los cubos de 1, 2, 3, 4 y 5 son 1, 8, 27, 64 y 125. Calculando las diferencias entre estos números se obtiene: 7 (8-1), 19 (27-8), 37 (64-27) y 61 (125-64). Estas son las diferencias primeras. Las diferencias segundas son 12, 18 y 24 (véase la figura adjunta). Todas las diferencias terceras valen 6, y lo mismo ocurre para la serie entera de los cubos, por mucho que prolonguemos la tabla. Por tanto, con saber tan sólo el valor del primer término de cada serie de diferencias es posible calcular toda la serie de cubos por simples sumas. Cada término será la suma del término anterior y la diferencia siguiente.




6
0

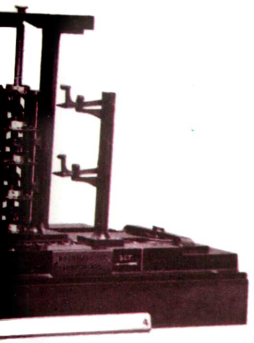


**La máquina de diferencias de Scheutz n.º 2.** Esta máquina fue utilizada hacia 1860.

Leibniz hizo otras dos aportaciones notables a la informática. La primera en 1666, cuando señaló que era posible analizar el razonamiento descomponiéndolo en una serie de elementos colocados en cierto orden. Esta idea está en la misma base del funcionamiento de los ordenadores. La segunda aportación tuvo lugar en 1679, cuando perfeccionó la teoría de los números binarios.

## Calculadoras eficaces

La primera máquina que alcanzó cierto éxito comercial fue el aritmómetro introducido en 1820 por el inventor francés Charles X. Thomas de Colmar. Se trataba esencialmente de una versión mejorada de la calculadora de Leibniz, incluyendo su rueda dentada. Sin embargo, el aritmómetro no empezó a venderse en grandes cantidades hasta 1860. Los avances posteriores tuvieron lugar principalmente en los Estados Unidos. El ingeniero George Barnard Grant fabricó una calculadora compacta «de cremallera y piñón». Más adelante, en 1875, Frank Baldwin inventó una rueda multiplicadora con nueve clavijas que entraban o salían, modificando el número de dientes. Dorr E. Felt puso a punto una calculadora en la que la acción de apretar teclas suministraba toda la energía necesaria para su funcionamiento. Construyó su pri-



mer modelo en 1885, bautizándolo con el nombre de «comptometer». Tuvo mucho éxito, lo mismo que la «Máquina de sumar e imprimir» de William S. Burroughs en 1888. Esta máquina era capaz no sólo de sumar números, sino también de imprimirlos todos ellos, más su suma, en una banda de papel.

**El padre del ordenador**

Estas máquinas de calcular no eran verdaderos ordenadores. Contaban con algún método para introducir números y obtener resultados, y disponían de mecanismos de cálculo que llevaban a cabo las operaciones aritméticas necesarias. Todo ello equivale a las unidades de entrada, salida y proceso de un ordenador, mas carecían de otros aspectos esenciales. No tenían memoria y además los mecanismos no realizaban más que operaciones aritméticas.

El primero en concebir un ordenador fue el inventor británico Charles Babbage. Babbage era matemático al tiempo que inventor, y utilizaba asiduamente los logaritmos para realizar los cálculos que su trabajo le exigía. Las tablas con las que se contaba entonces contenían errores, y Babbage, en 1812, concibió la idea de una máquina capaz de compilar tablas de logaritmos exactas. La máquina que diseñó se basaba en el principio llamado método de las diferencias. Para 1821 contaba ya con un prototipo que confirmaba la validez de sus ideas. En 1823 el gobierno británico concedió a Babbage una primera subvención, y se inició la construcción de la Máquina de Diferencias.

La Máquina de Diferencias era una inmensa colección de engranajes interconectados que habían de fabricarse con tolerancias muy finas, más de lo que la ingeniería de la época era capaz de producir. La escasa precisión de los engranajes se acumulaba de tal modo que la Máquina experimentaba sacudidas y vibraciones atroces, incluso tras ser rediseñada un sinnúmero de veces. El gobierno siguió financiando a Babbage durante diez

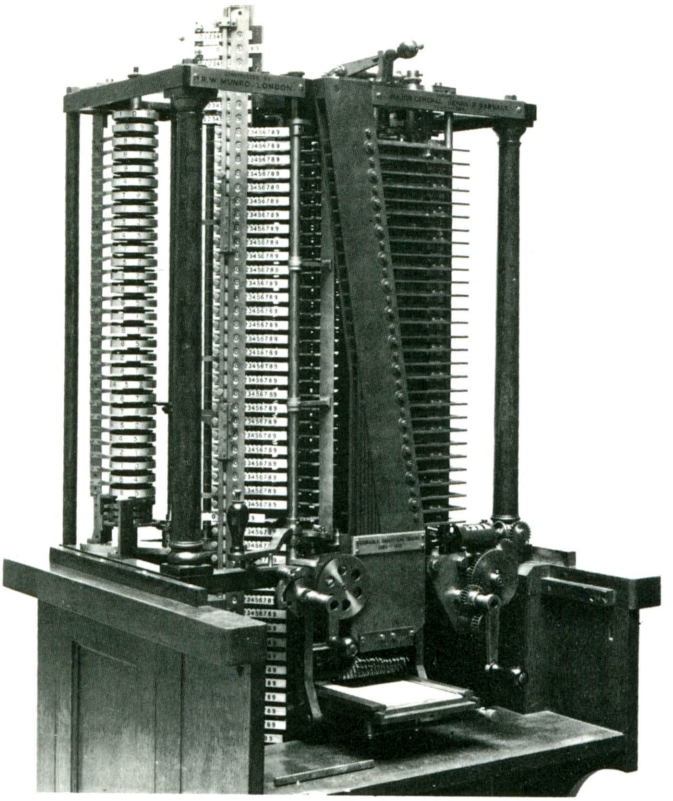
años, invirtiendo una fuerte suma de dinero, hasta que por fin le retiró su apoyo. La Máquina llegó a un punto muerto. Sin embargo, lo fundamental de su diseño no tenía error, y en manos de una ingeniería más avanzada acabaría por funcionar.

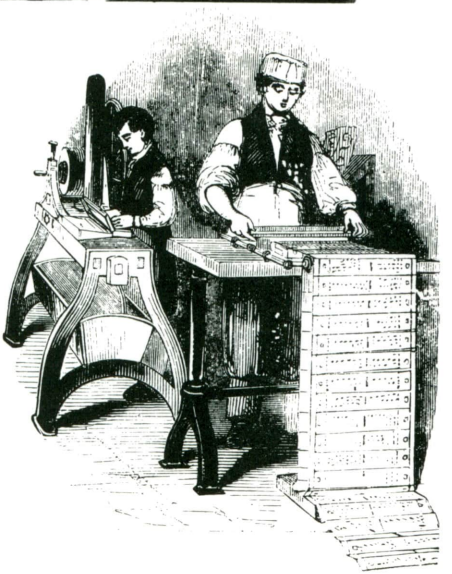
En 1834, un año después de retirada la Máquina, el ingeniero sueco George Scheutz tropezó con un informe que hablaba de ella y decidió construir una. Scheutz rediseñó algunas partes y terminó su máquina en 1840. Después procedería a perfeccionarla, recibiendo el apoyo del gobierno sueco.

A Babbage pudo haberle desmoralizado esta situación, pero no se arredó. En medio de su lucha con las complicaciones de la Máquina de Diferencias, le vino a la cabeza una idea aún más ambiciosa: la construcción de una máquina de calcular polivalente, que funcionara de maneras distintas según el tipo de problema que se quisiera resolver. En este momento, hacia 1833, la idea del ordenador había nacido. Desde el punto de vista de su factibilidad, estaba más por delante de su tiempo aún que la Máquina de Diferencias.

Babbage llamaría a esta nueva invención Máquina Analítica. Para introducir datos en la Máquina y darle las instrucciones necesarias para que llevase a cabo un determinado cálculo —es decir un programa— propuso la utilización de tarjetas perforadas como las ideadas por el inventor francés Joseph Jacquard en 1801. Estas tarjetas se aplicaban en un principio al control del funcionamiento de los telares, para tejer distintos dibujos.

Las operaciones de cálculo las realizaría un mecanismo aritmético que Babbage llamaba «el taller», y que contaba con una unidad de control para hacerle seguir las instrucciones. Los resultados temporales se transferirían al «almacén», que constaría de mil columnas de cincuenta ruedas cada una. En la Máquina Analítica encontramos ya los rasgos básicos de un ordenador. Unidades de entrada y salida para introducir datos e instrucciones y para imprimir resultados; una unidad de pro-





**La Máquina Analítica de Babbage.** Babbage, matemático e inventor británico, comenzó a diseñarla en 1833. Pensaba utilizar tarjetas perforadas como los telares Jacquard (derecha). Por desgracia, sólo llegó a existir un prototipo.

ceso completa, el taller, con una unidad de control para hacerle ejecutar un programa; y una unidad de memoria, el almacén.

Sin embargo, sólo llegarían a construirse algunas partes sueltas de este prototipo de ordenador. La máquina completa estaba absolutamente fuera del alcance de la técnica de la época. Nadie aceptó el desafío, como había ocurrido con la Máquina de Diferencias, y Babbage murió en 1871, dejando en herencia un concepto maravilloso.

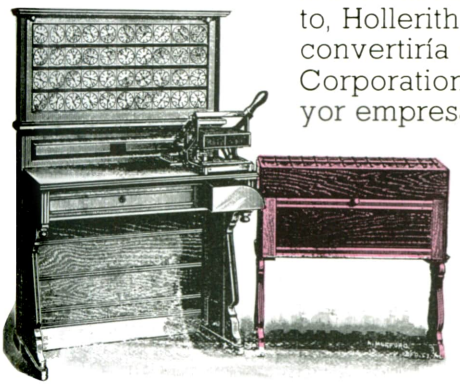
**Llega el siglo XX**

Aunque durante un siglo no se progresaría más hacia el ordenador, tuvieron lugar diversos avances en otras direcciones. El ordenador de Babbage era digital: sus resultados se hubieran podido imprimir en forma de columnas de números. Era un ordenador de los que, en versión electrónica, usamos hoy. Pero existe otro tipo de ordenador, el analógico, cuyos resultados no son números, sino cierta cantidad que representa un número. Los mecanismos engranados que Babbage quería usar eran ideales para desarrollar un ordenador analógico, porque permitían visualizar los resultados en forma de gráficos. La primera máquina analógica de este tipo la construyeron el científico británico lord Kelvin y su hermano James Thomas en 1872. Se utilizaba únicamente para predecir las pleamares y las bajamares, y representar gráficamente sus niveles.

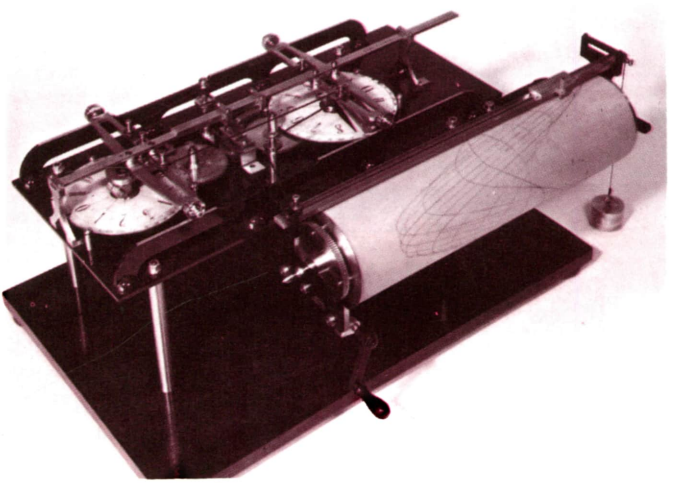
Kelvin pensaba que sería posible construir un ordenador analógico que resolviera cualquier tipo de problema, pero por dificultades técnicas nunca trató de diseñarlo ni de construirlo. Lo que sí hizo fue darle un nombre, analizador diferencial (podría resolver cualquier ecuación diferencial). La primera máquina de este tipo la construiría en Norteamérica Vannevar Bush en 1930.

Mientras tanto se habían registrado grandes progresos en otro terreno de interés, el de la manipulación de la información para extraer de ella resultados significativos, lo que llamamos «proce-

so de datos». Los provocó la necesidad de analizar los censos norteamericanos de finales del pasado siglo. Se hizo necesario un método más eficaz que la consignación y el recuento manuales, y se estableció un concurso para contrastar las propuestas de nuevos sistemas. Lo ganó Herman Hollerith con su máquina tabuladora. Se le escogió para trabajar con el censo de 1890, y tan sólo seis semanas después del día del censo Hollerith pudo anunciar que la población de los Estados Unidos era ligeramente superior a 62 millones de personas. El recuento del censo anterior de 1880 había durado siete años. Respaldado por este éxito, Hollerith fundó una compañía que en 1924 se convertiría en la International Business Machine Corporation o IBM, que en la actualidad es la mayor empresa de informática del mundo.



**El tabulador de Hollerith.** El norteamericano calculó el censo de 1890 de su país con su tabulador de tarjetas perforadas.



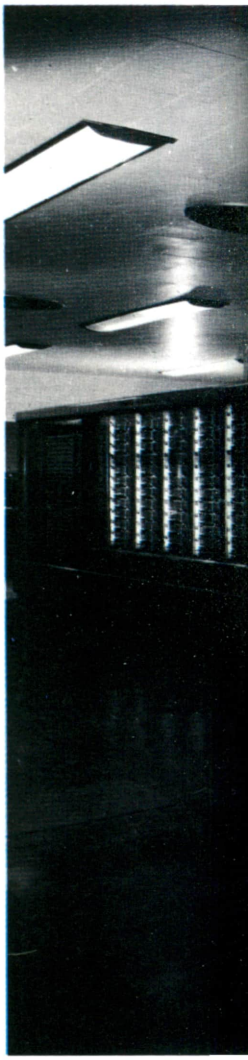
**La máquina de predecir mareas de Kelvin.** Construida por J. White en 1872. La máquina de Kelvin las predecía trazando unos gráficos. La cuantificación representando valores numéricos es la característica principal del cálculo analógico.

## Los primeros ordenadores eléctricos

Los Estados Unidos estaban por entonces en situación de efectuar un avance auténticamente revolucionario. IBM poseía los recursos tecnológicos y Bush había inspirado importantes avances teóricos. Su alumno Claude Shannon determinó los circuitos necesarios para realizar aritmética binaria en 1937, y George Stibitz, colega de Shannon, construyó los primeros modelos operativos basados en relés electromagnéticos. Stibitz comprendió que estos circuitos podían ser la base de un ordenador programable y diseñó uno que nunca se llegó a construir, pero en 1940 concluyó una unidad experimental llamada *Complex Number Computer,* capaz de trabajar en binario.

Bajo la dirección de Howard Aiken, IBM concentraba sus esfuerzos en la fabricación de un ordenador auténtico. Los trabajos comenzaron en 1939 y finalizaron en 1944 en la Universidad de Harvard. El nombre de la máquina sería Harvard Mark I y, aunque lenta en sus cálculos, cabe considerarla el primer ordenador digital.

Años atrás, en 1938, un ingeniero alemán llamado Konrad Zuse había construido en su casa una unidad experimental que poseía todas las características de un ordenador binario digital, el Z1. Su unidad de entrada era un teclado, un conjunto de conmutadores la memoria, y una hilera de luces daba los resultados binarios. Zuse crearía después el Z2, con relés y entrada por película perforada. Durante la Segunda Guerra Mundial se sucedieron el Z3 y el Z4, máquinas electromecánicas que se usaron en el diseño de aviones. Todas estas máquinas quedaron destruidas en la guerra.



## Los primeros ordenadores electrónicos

Los tubos o válvulas electrónicas se habían inventado a finales del siglo pasado. Las válvulas tienen la propiedad de conectar y desconectar rápidamente una corriente eléctrica, y por ello se podían emplear en los ordenadores para manipular datos codificados en pulsos eléctricos. Aiken lo

sabía cuando diseñó el Harvard Mark I, pero prefirió los relés porque, aunque lentos, eran más fiables. Las válvulas, grandes consumidoras de energía, se sobrecalentaban, descomponiéndose a menudo. También el gobierno británico era consciente de estos factores. El científico Alan Turing y otros se aprestaron a construir un ordenador ca-

paz de descifrar los mensajes en clave del enemigo. Sus máquinas, construidas a partir de 1941, eran de relés, siendo sustituidas en 1943 por los ordenadores de válvulas de la serie Colossus. Estas máquinas, tenidas en el mayor de los secretos, descifraron enseguida los códigos alemanes.

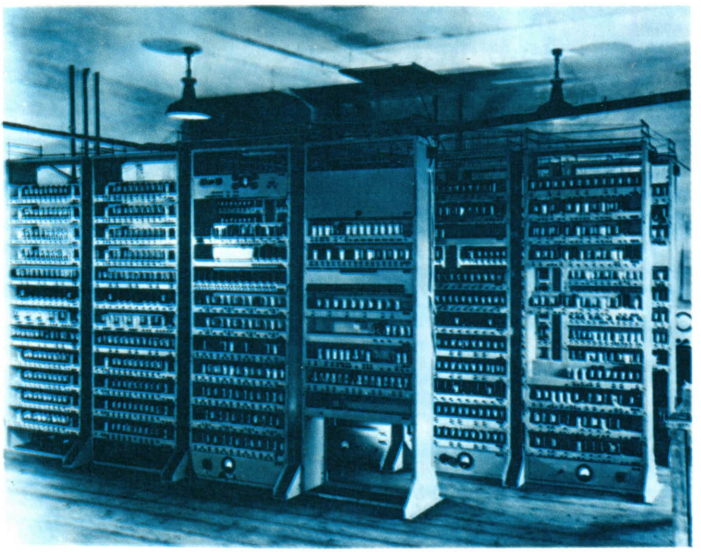
El Colossus fue el primer ordenador electrónico digital, pero sólo podía realizar una tarea, descifrar códigos. Mientras, en Estados Unidos empezaba a tomar forma otro ordenador electrónico, creación de John W. Mauchly y J. Presper Eckert. Mauchly y Eckert emprendieron la construcción de un ordenador de válvulas para utilizarlo en cálculos balísticos, y no concluyeron hasta 1946. El ordenador se llamó ENIAC *(Electronic Numerical Integrator And Computer)* o Integrador y Calculador Numérico Electrónico. ENIAC era asombrosamente rápido, pudiendo sumar un número de cinco cifras cinco mil veces en un segundo, usando aritmética decimal y no binaria. Era ligeramente más rápido que el Colossus y, aunque contaba con poca memoria, se le podía programar para llevar a cabo distintos cálculos. El proceso era laborioso, pues significaba rehacer conexiones y cableados, pero ENIAC fue el primer ordenador electrónico digital polivalente. Estuvo diez años en servicio.

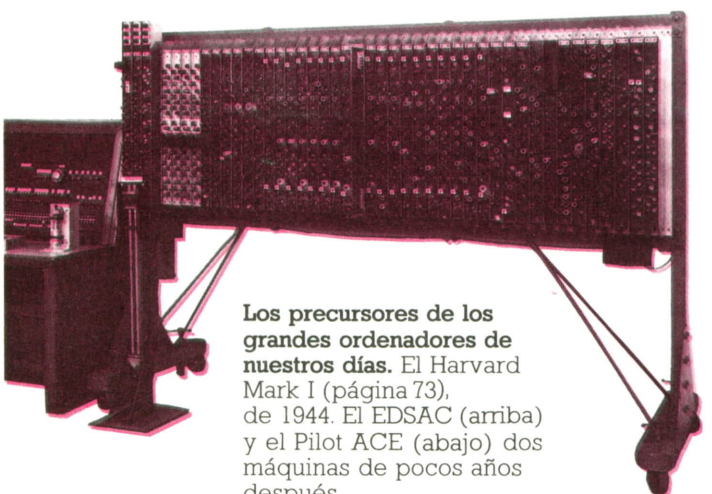
**El desarrollo de los ordenadores hasta nuestros días**

Británicos y norteamericanos encabezaban por entonces el desarrollo de los ordenadores a nivel mundial; sin embargo, dos pasos cruciales en la evolución de los ordenadores se dieron en Estados Unidos.

El primero lo dio el matemático John von Neumann. Comprendiendo enseguida que las posibilidades de ENIAC quedaban muy recortadas por lo difícil que resultaba programarlo, sugirió almacenar en la memoria del ordenador los programas en forma de instrucciones codificadas. Así, el ordenador podría tener acceso instantáneo al programa e interactuar con él si fuera necesario. El

primer ordenador donde se ejecutó un programa almacenado en memoria fue el Manchester Mark I, ordenador británico que comenzó a funcionar en 1948. Le siguió en el año 1949 el EDSAC *(Electronic Delay Storage Automatic Calculator)*, construido en Cambridge, Inglaterra, por un equipo dirigido por Maurice Wilkes.





**Los precursores de los grandes ordenadores de nuestros días.** El Harvard Mark I (página 73), de 1944. El EDSAC (arriba) y el Pilot ACE (abajo) dos máquinas de pocos años después.

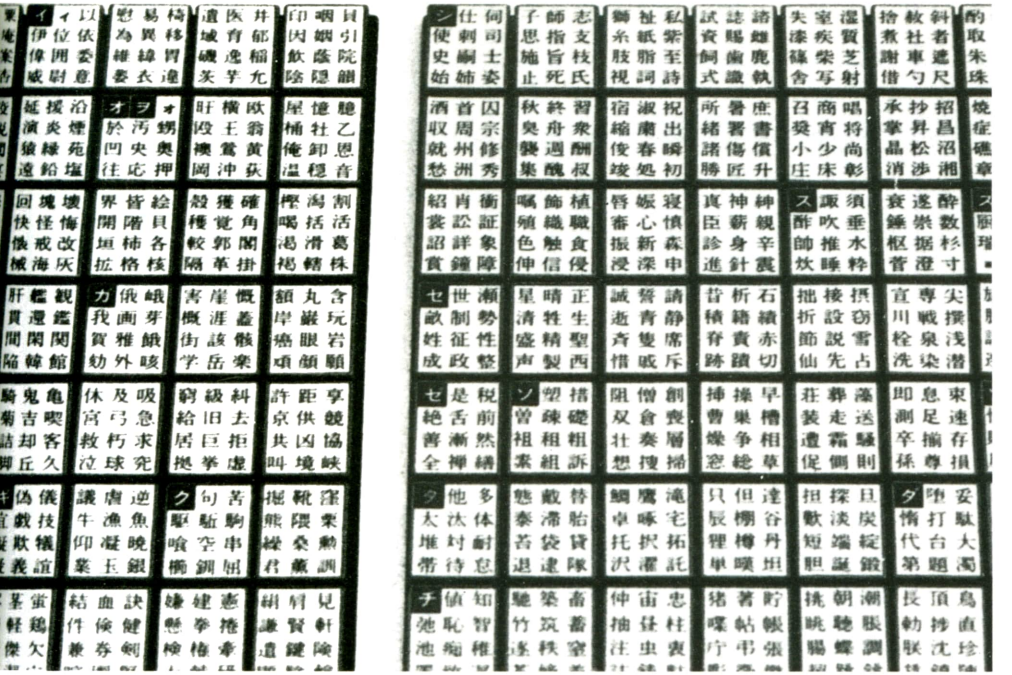
Estos primeros ordenadores estaban destinados a fines científicos o militares, pero el terreno comercial atraía a muchos, además de a Mauchly y Eckert. Su nuevo ordenador, BINAC, fabricado en 1948, utilizaba cintas magnéticas como dispositivo de entrada, siendo más rápido y barato que ENIAC. Ambos técnicos desarrollaron luego el primer ordenador comercial, el UNIVAC I, para Sperry Rand, en 1951. Su capacidad era cien veces mayor a la de ENIAC, y diez veces más rápido, pero en tamaño casi diez veces menor.

En este momento los Estados Unidos comenzaron a tomar la delantera en el desarrollo de los ordenadores. IBM empezaría a dominar el mercado informático con la serie 700 de ordenadores de válvulas, aparecida en 1952. Eran máquinas difíciles de operar y muy caras, lo que limitaba su mercado potencial a los departamentos ministeriales y grandes empresas.

Este obstáculo al ulterior desarrollo de la informática quedó superado gracias al segundo de los pasos cruciales antes mencionados: la invención del transistor, que tuvo lugar en Estados Unidos en 1948. Fue el resultado de largas investigaciones realizadas por John Bardeen, Walter Brattain y William Shockley , que compartirían el premio Nobel concedido a su hazaña. Los transistores hacían lo mismo que las válvulas pero en un espacio más pequeño y con un consumo de energía eléctrica menor.

El transistor no quedó incorporado inmediatamente a los ordenadores. Primero hubo de perfeccionarse hasta que su rendimiento igualase al de las válvulas. Sucedió en 1956 con el primer ordenador transistorizado, el TX-O *(Transistor Experimental)* en el Instituto de Tecnología de Massachusetts. Esta máquina fue la precursora de una auténtica revolución en el diseño de ordenadores: en el plazo de cuatro años todos los ordenadores usaban transistores. De un golpe se redujeron tamaño y precio, y se aumentaron potencia y sencillez. Además mejoró la memoria en fiabilidad y

**Teclado japonés.** El uso de procesadores de textos se ha generalizado en Occidente, pero los países cuyo alfabeto no es el romano padecen graves problemas a la hora de automatizar. A japoneses y chinos les resulta difícil representar electrónicamente sus caracteres. Mayor problema aún es teclearlos. A la derecha una solución japonesa: un teclado de doce caracteres por tecla. En China se fomenta un alfabeto romanizado.



capacidad con el desarrollo de sistemas de discos y cintas magnéticas, con lo que los ordenadores empezaron a ser accesibles a industrias y comercios de toda índole.

El salto cualitativo que introdujo el transistor en el diseño de ordenadores fue sólo el comienzo de un proceso cada vez más acelerado de miniaturización de sus componentes, con la consiguiente reducción de precios y aumento de potencia. Y si un componente tal como el circuito que hace de válvula podía comprimirse en el pequeño trozo de semiconductor que constituía un transistor, lo mismo pasaba con todo un grupo de componentes. Este concepto de circuito integrado se aireó por vez primera en 1952, pero no se lograría su operatividad hasta 1958, año en que fue desarrollado por Jack Kilby, de Texas Instruments. Su perfeccionamiento duró varios años, y los primeros ordenadores con circuitos integrados empezaron

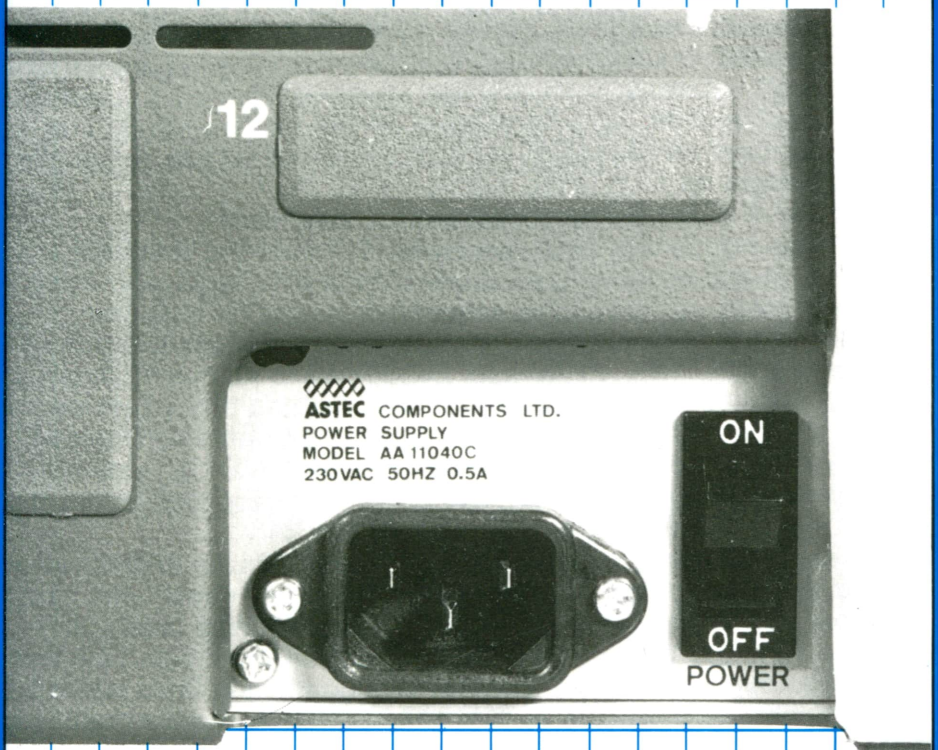
## TEORIA INFORMATICA

Siguiendo las investigaciones de Leibniz sobre los números binarios y la naturaleza del razonamiento, George Boole analizó a mitad del s. XIX los procesos lógicos y demostró que se podía abordar matemáticamente la lógica, utilizando símbolos como notación algebraica para denotar operaciones tales como «sí» u «o». Con este método, llamado álgebra de Boole, podía averiguarse el resultado lógico de una serie de proposiciones. A. Whitehead y B. Russell desarrollaron la matemática de la lógica, que condujo al diseño de circuitos eléctricos capaces de llevar a cabo operaciones lógicas comparando números, además de realizar operaciones aritméticas.

a aparecer en Estados Unidos en 1964. Por entonces se podían comprimir treinta componentes en un *chip* de cinco milímetros de lado; a los diez años la cifra era mil veces mayor y no ha dejado de crecer. Uno de los resultados de esta tecnología del *microchip* ha sido la aparición del microordenador. Con esto se llega a la cuarta generación de ordenadores, figurando en la primera los de válvulas, en la segunda los de transistores y en la tercera los de circuitos integrados.

Todos estos avances tuvieron lugar en Estados Unidos, donde también se desarrolló el proyecto MAC, primer sistema en el que se interconectaron ordenadores a una red. Sucedió a principios de los años 60, y fue un paso muy importante en la potenciación de los ordenadores. También lo había sido la aparición de lenguajes de ordenador para facilitar la programación, desarrollados principalmente en Norteamérica. Sin embargo, al acercarnos a la quinta generación de ordenadores, el predominio estadounidense parece amenazado. Tal vuelco habría que atribuirlo al hecho de que un país ha convertido la consecución del ordenador de la quinta generación en una especie de objetivo nacional: Japón. Su fecha límite: 1990.

# LOS PRIMEROS PASOS



# LA PUESTA EN MARCHA

Una vez decididos a comprar nuestro primer ordenador, ¿qué hacer? Abierta la caja que contiene la máquina, encontramos una confusa colección de cajas negras y cables que parecen extenderse por todas partes sin finalidad aparente. Lo que es el ordenador en sí, motivo al fin y al cabo de la compra, está como perdido en la barahúnda.

**Desempaquetado del ordenador**

No desesperemos. Todo lo que contiene la caja sirve para algo. Vale más sacarlo todo e irlo poniendo en una mesa delante de nosotros.

Comprobemos cuidadosamente que no nos falta nada con ayuda de la documentación que acompaña al equipo. En una típica lista de comprobación figurarán los siguientes elementos:

1 La consola del ordenador. Es la parte más importante del equipo y posiblemente llevará incluido el teclado.

2 Una fuente de alimentación. La mayor parte de los ordenadores no pueden conectarse directamente a la red, siendo preciso interponer una fuente de alimentación suplementaria. Esta unidad suele calentarse mucho cuando lleva cierto tiempo encendida y funcionando, pero a menos que al tocarla notemos que verdaderamente quema no hay que preocuparse de ello.

3 Prácticamente todos los ordenadores vienen

con diagramas que ilustran cómo hay que conectar la máquina a la pantalla, en nuestro caso probablemente el televisor de la casa.

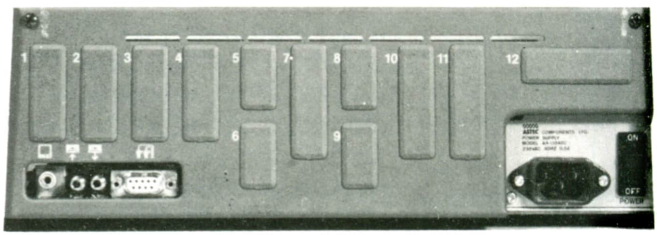
4 Cables. Suponiendo que todo esté bien conectado, ya podemos enchufar la fuente de alimentación al conector correspondiente del panel posterior del ordenador, y el otro extremo a la red. No hay que encender todavía, porque quedan por hacer algunas comprobaciones.

5 Un modulador de RF. Este dispositivo convierte la señal de vídeo producida por el ordenador en una señal susceptible de aparecer en un televisor. No todos los ordenadores, sin embargo, lo precisarán, pues algunas máquinas producen una señal que se puede emitir directamente al televisor. Un extremo del modulador de RF, si existe, se enchufa en la parte trasera del ordenador, en el conector que lleva los rótulos «RF-in» o «antenna». El otro extremo se enchufa en el conector de la antena que hay en la parte de atrás del televisor. Sin este modulador algunas máquinas no tendrían posibilidad de salida por pantalla. Si en lugar de un televisor se usa un monitor, llamado a veces tubo de rayos catódicos (CTR o *cathode ray tube)* o VDU *(visual display unit)* seguro que no necesitamos modulador de RF. Hay que tener presente, sin embargo, que muchos monitores, especialmente los más baratos, no reproducen imágenes en colores, ni sonidos. Estas son dos de las características más deseables de la mayoría de los ordenadores domésticos, y carecer de ellas puede limitar gravemente tanto la gama de *software* utilizable como, lo que es peor, la diversión que el ordenador puede proporcionarnos. Si podemos permitirnos el gasto que supone adquirir un monitor sofisticado con sonido y color, la calidad de la imagen producida será muy superior a la que daría un televisor. Esto se debe a que el proceso de conversión que lleva a cabo el modulador de RF provoca una ligera degradación en la imagen.

El equipo que hemos descrito constituirá normalmente cuanto necesitamos para conectar

## MONTAJE DEL ORDENADOR

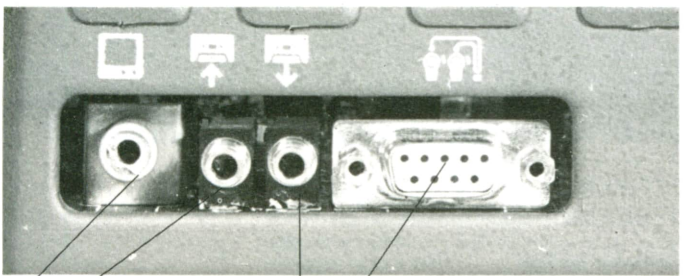
Tras sacar la máquina de la caja, lo probable es que encontremos una serie de componentes y una maraña de cables. El manual dirá en qué conector va cada cable, pero no siempre resulta fácil descifrar la documentación. En realidad las cosas no son tan complicadas. Primero, todos




**El Apple IIe.** Ha sido diseñado para facilitar al máximo su montaje. Todos los conectores están agrupados en la parte posterior de la máquina (figura de arriba a la derecha). Una de las características del Apple es su capacidad de expansión —dentro de la máquina hay diversos conectores que permiten añadir una amplia gama de tarjetas que proporcionan más memoria y nuevas posibilidades de entrada-salida. A lo largo de la parte posterior de la máquina hay diversos paneles desmontables sin uso definido (son visibles en la foto de arriba), en cuyo lugar pueden situarse los conectores de estos nuevos puertos.

nuestro nuevo ordenador, encenderlo, y ver cómo comienzan a aparecer en la pantalla los frutos de nuestros esfuerzos. Sin embargo, sólo con hacer esto no llegaremos muy lejos. Aunque nos limitemos a recorrer los ejemplos de programas que vengan en nuestro manual del usuario, cada vez que apaguemos la máquina se perderá todo lo que hubiera en pantalla, se borrará completamente de la memoria del ordenador (véase Módulo 19). En otras palabras, no contamos aún con ningún medio de «guardar» lo que hemos tecleado. Para superar esta dificultad, todos los ordenadores cuentan con la posibilidad de conectarse, me-

los elementos deben estar ligados de algún modo a la fuente de alimentación. Muchos micros cuentan con un transformador externo que se enchufa al ordenador y a la red. Para que el ordenador pueda mostrar una imagen en la pantalla de un televisor, conéctese el cable de vídeo al conector para la antena que tiene el televisor y a la clavija apropiada del ordenador. El magnetófono se conectará al ordenador, uniendo las clavijas con el rótulo EAR tanto en uno como en otro y, también, las rotuladas MIC.



**Conector de salida de datos.** Salida de datos y conecta al MIC.

**Mandos para juegos.** Permite conectar hasta dos mandos para juegos.

**Conector de vídeo.** Permite la salida a un monitor y una *interface* opcional para televisor.

**Conector de entrada de datos.** Para transferir al micro datos desde el magnetófono, debiéndose conectar a la clavija EAR de éste.

diante una *interface*, a algún tipo de dispositivo capaz de almacenar la información que le hemos introducido. Hay dos tipos de medios de almacenamiento. El más corriente es el magnetófono ordinario, que resulta satisfactorio para la inmensa mayoría de las aplicaciones domésticas. El otro dispositivo de almacenamiento habitual es el disco magnético, pero por el momento nos centraremos exclusivamente en la *cassette*.

## Compatibilidad de los equipos

Topamos aquí con la primera división importante entre marcas de ordenadores, y merece la pena

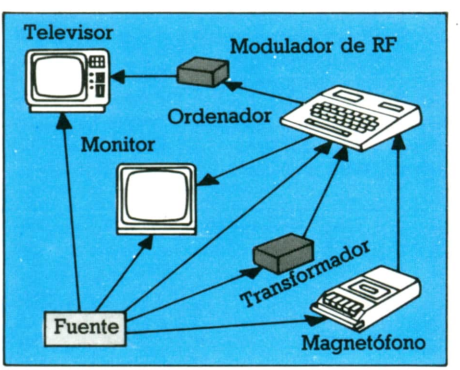
**El Spectrum.** Es fácil tanto su montaje como su utilización. Todos los conectores están agrupados en la parte de atrás. Carece de posibilidades de expansión interna, permitiendo un conector grande, situado en la parte posterior, la incorporación de otros periféricos. Con este conector se tiene acceso a un gran número de señales generadas por el ordenador.

**Fuente de alimentación.** Hay un transformador externo para reducir el voltaje de la red.

**Conector de MIC.** Salida del ordenador al magnetófono.

**Conector de entrada.** El ordenador recibe los datos a través de él. Unase a la clavija similar del magnetófono.

**Modulador de vídeo.** Permite conectarle un televisor.




**Montaje típico.** Ordenador doméstico típico conectado a la fuente de alimentación y a distintos periféricos. No precisaremos de un transformador si nuestra máquina lo lleva internamente. Tampoco un modulador de RF si usamos un monitor en vez de un televisor.

señalar esta diferencia. Por lo que se refiere al almacenamiento en cinta, los ordenadores se dividen en dos clases, los Commodore y los demás.

La firma Commodore tomó la decisión de fabricar ordenadores que no pueden conectarse a los magnetófonos corrientes (véase Módulo 21), siendo imprescindible adquirir uno especial que sólo

vende Commodore. En el Módulo 9, que trata de cómo guardar y recuperar programas y datos en general, examinaremos más a fondo esta complicación, pero si hemos comprado una de estas máquinas hay una cosa que nos interesa en este momento. Si tenemos un ordenador Commodore, la primera vez que lo desempaquetamos, la unidad de cinta especial lleva un cable incorporado que se enchufa en la parte trasera de la consola del ordenador. En otras marcas el cable va suelto, con lo que puede usarse casi cualquier magnetófono.

Para que el magnetófono pueda «hablar» al ordenador y también a la inversa, lo normal es que haya dos cables (véase página anterior). Un cable va del ordenador al magnetófono y el otro a la inversa. No hace falta más que enchufarlos (cerciorándose de que todos los conectores estén bien apretados) y casi estamos listos para empezar.

Recapitulando pues, hasta ahora hemos conectado el ordenador a nuestro televisor (o quizás a un monitor), tal vez con ayuda de un modulador de RF, enchufado el ordenador a la red (a través de una unidad de fuente de alimentación) y, salvo en el caso del Commodore, conectado también un magnetófono para utilizarlo como medio de almacenamiento. Los propietarios de un Commodore habrán enchufado su unidad de cinta especial.

**Sintonización**

A los que no usen un monitor, el ajustar su televisor hasta obtener una imagen buena y nítida puede llevarles algo de tiempo. Encenderemos tanto el ordenador como el televisor y sintonizaremos éste al canal recomendado en el manual del usuario. Obviamente, si tenemos varios canales libres, cada uno con su mando de ajuste independiente, es preferible usar uno de ellos y no uno de los que solemos usar. La mayor parte de los ordenadores domésticos muestran en pantalla al encenderlos un logotipo o un aviso sobre los derechos de propiedad o *copyright*, y esto es lo que hay que buscar a la hora de ajustar el mando.

Obtener el mensaje inicial de *copyright* suele ser fácil, pero muchos ordenadores muestran sólo una imagen en blanco y negro al ser encendidos, aunque sean capaces de generar color. Para cerciorarnos de que hemos conseguido una perfecta sintonía hay que cambiar de color. En el ahora tan popular Spectrum de Sinclair, por ejemplo, basta pulsar primero la tecla CAPS SHIFT (extremo inferior izquierdo) y luego, sin soltarla, la tecla BORDER (hacia la mitad de la fila inferior de teclas). Aparecerá en pantalla la palabra BORDER. Pulsamos luego la tecla BLUE (el número «1») en el extremo izquierdo de la fila superior. Por último, pulsamos la tecla ENTER, que se encuentra a la derecha del teclado. Si tenemos correctamente sintonizado el televisor, los bordes de la pantalla se pondrán azules. Si, por el contrario, salen negros, hay que seguir ajustando el mando hasta conseguir un buen color azul oscuro en la pantalla.

En ordenadores de otras marcas las instrucciones de color serán distintas de las mencionadas. No hay que preocuparse demasiado si no se entiende cómo funcionan en un primer momento. Podemos «cargar» (transferir a la memoria del ordenador desde un medio de almacenamiento) la cinta de demostración que probablemente acompaña al ordenador, y si con ello no se obtiene una imagen en color se puede corregir la sintonía mientras está ejecutándose el programa de demostración. O sencillamente dejarlo para cuando conozcamos mejor nuestra máquina concreta y estemos más familiarizados con las instrucciones que gobiernan el color.

Por cierto, si nuestro televisor tiene control de frecuencia automático debemos suprimirlo mientras hacemos esta operación de sintonizarlo. Y sobra decir que nuestro ordenador sólo producirá una imagen en color si lo conectamos a un televisor de color. Es sorprendente, pero muchas personas creen que se puede sacar una imagen en color en un televisor de blanco y negro.

Si, después de todo esto, seguimos recibiendo

una imagen de poca calidad, o no vemos nada en absoluto, tratemos de sintonizar otro canal hasta dar con uno que sea adecuado. Si tampoco lo logramos, lo mejor es preguntar en la tienda donde nos lo vendieron. Sin embargo, muy pocas veces ocurre esto, y conviene buscar antes otras causas posibles. Volvamos a mirar en el panel trasero del ordenador, comprobando que todos los cables están bien puestos; también podría haberse aflojado el conector del televisor.

Si nos paramos a inspeccionar nuestro ordenador recién montado, lo más probable es que veamos un laberinto de cables encima de la mesa y otros esparcidos por el suelo, como esperando a que alguien pase y tropiece con ellos. Así que conviene dedicar un ratito a ordenarlo todo, poner los cables donde no estorben y despejar la mesa de todo lo que no sea el equipo absolutamente esencial para trabajar.

# FAMILIARIZACION

Una vez que está todo correctamente conectado, y tenemos el ordenador encendido y en marcha, ¿qué hacer? Lo más probable es que nos hayamos quedado mirando la pantalla del televisor, en la que aparece el mensaje de *copyright* del ordenador y poco más, preguntándonos cómo demonios se usa el equipo que acabamos de comprar.

**Empecemos por conocer el teclado**

A estas alturas ya nos habremos dado cuenta de que el teclado del ordenador es el que permite controlarlo todo. Los teclados, como los ordenadores, pueden adoptar muchas formas, pero hay dos (con ligeras variantes) que parecen dominar actualmente el mercado del ordenador doméstico. Una de las más comunes, aunque sea por el mero hecho de llevarla el ordenador doméstico más vendido, es la del ZX Spectrum de Sinclair.

El Spectrum tiene un teclado pequeñísimo, por lo que se hace incómodo. Este tipo de teclado no es «sensible al tacto»; en otras palabras, hay que apretar la tecla para obtener en pantalla una respuesta, no basta tocarla. Los teclados sensibles al tacto son totalmente planos, y llevan en la superficie los nombres de las teclas. Debajo tienen unos contactos eléctricos que transmiten las señales al resto del sistema. Estos sistemas resultan más baratos de fabricar, pero no tan cómodos de usar.

Una característica común a todos los teclados pequeños es el llamado método para introducir datos «por pulsación única». «Dato» es un término al que conviene irse habituando, pues todos los iniciados en informática lo utilizan para referirse a las palabras y la información. La entrada por pulsación única compensa en cierta forma el bajo número de pulsaciones que se consigue en los teclados de este tipo. Con sólo pulsar una o, más frecuentemente, dos teclas, puede obtenerse en la pantalla una palabra (u orden) completa. Estas palabras no son términos corrientes; son palabras que el ordenador «entiende» y sabe traducir.

Tomemos por ejemplo la palabra PRINT. Significa y hace exactamente lo que sería de esperar, es decir, imprime algo. Con el pequeño teclado del Spectrum se puede introducir esta palabra sin más que pulsar una tecla. En casi todos los demás ordenadores habría que pulsar cinco (P-R-I-N-T) para conseguir el mismo resultado. Como esta palabra en particular se usa tanto, también otros ordenadores tienen un método abreviado de introducirla.

Así pues, en los teclados pequeños resulta muy fácil escribir palabras inteligibles para el ordenador, pero el verdadero tecleo (por ejemplo, cuando se utiliza el ordenador en tratamiento de textos, véase Módulo 36) es un tanto lento.

En los teclados de pulsación única notaremos que cada tecla contiene hasta cinco o seis letras, símbolos o palabras enteras, al estilo de ciertas calculadoras. Se trata sencillamente de un modo de reducir el tamaño físico del teclado. Esta concentración de efectos en la misma tecla puede llegar a ser irritante. Vamos a teclear una cosa y resulta que estamos en un «modo» en el que no se puede introducir información de ese tipo y tenemos que pulsar primero otras teclas antes de seguir o incluso volver atrás y empezar de nuevo. Sin embargo, pasado cierto tiempo termina uno por acostumbrarse a este sistema.

En el resto de las máquinas, por el contrario, en-

contramos normalmente un teclado ortodoxo, en el que cualquier mecanógrafa competente se sentirá enseguida a sus anchas. Evidentemente, algunos de los nombres de las teclas serán exclusivos de nuestra máquina concreta. Probemos el teclado de nuestro ordenador. No importa que lo que tecleemos sean perfectas tonterías, porque esto no puede hacerle ningún daño a la máquina; se trata simplemente de familiarizarse con la distribución de las teclas y con sus funciones.

**Uso de cada tecla**

Dos teclas que adquieren una importancia crucial cuando empezamos a utilizar nuestra máquina son la INSERT y la DELETE. Como cabe imaginar, sirven para insertar y borrar caracteres. Algunos teclados llevan teclas de funciones especiales que, cuando tengamos más experiencia, podremos programar para generar órdenes complejas o de uso muy corriente con sólo pulsar una de ellas.

La única forma de familiarizarse con el teclado es usarlo, así que propondremos un pequeño programa que el lector pueda teclear para ver qué sucede. El lenguaje que usaremos para comunicarnos con nuestro ordenador será, casi con seguridad, el Basic (que significa Código de Instrucciones Simbólicas de Usos Múltiples para Principiantes o *Beginner's All-purpose Symbolic Instruction Code).* Lamentablemente, las diferentes máquinas utilizan versiones ligeramente distintas del Basic, pero existen muchísimas probabilidades de que el siguiente programa sea compatible con el particular «dialecto» del Basic de nuestra máquina. Por tanto, tecleemos las siguientes líneas:

```
10 PRINT "¿COMO TE LLAMAS?"
20 INPUT A$
30 PRINT "HOLA";A$
40 FOR I=1 TO 100
50 PRINT A$
60 NEXT I
```

¿Qué son esos números que hay al principio de cada línea? Los ordenadores son máquinas «en blanco», y sólo pueden seguir nuestras instrucciones si se las presentamos en el orden debido. Los números de línea especifican precisamente ese orden. Aumentan de diez en diez por si queremos volver atrás en algún momento y añadir líneas nuevas. Esto podría deberse a que hubiéramos cometido un error o deseásemos alterar el resultado final del programa.

Lo que hace el programa, línea por línea, es preguntar por nuestro nombre y esperar a que se lo tecleemos. Una vez tecleado tenemos que pulsar la tecla ENTER (o RETURN) para que el ordenador pueda seguir adelante —él no sabe qué hacer si no se lo decimos—. A continuación escribirá HOLA y, detrás, nuestro nombre, pasando luego a escribir dicho nombre en la pantalla cien veces.

Es probable que para introducir este programa hayamos tenido que pulsar teclas que nunca antes habíamos tocado, y el siguiente nos dará oportunidad de ensayar unas cuantas más:

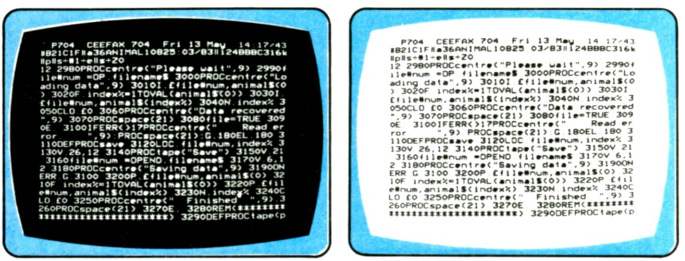
```
10 A$="YO SOY UN ORDENADOR"
20 PRINT A$
30 B$="Y TU ERES SOLAMENTE UN SER HUMANO"
40 PRINT B$
```

Podemos intentar modificar ambos programas nosotros mismos cambiando algunas de las palabras o frases.

Los programitas de este tipo, aunque son didácticos, no aprovechan realmente las posibilidades del ordenador. Más avanzado el libro se proporcionan orientaciones para redactar programas más complejos (véase Módulo 27). Como ya se ha dicho, hay muchas variedades de Basic distintas, y por eso no pueden darse más que orientaciones. Aunque este Basic corrientísimo será compatible con el 99% de los ordenadores domésticos, hay una notable excepción, el Jupiter Ace. Este ordenador utiliza un lenguaje totalmente distinto, lla-

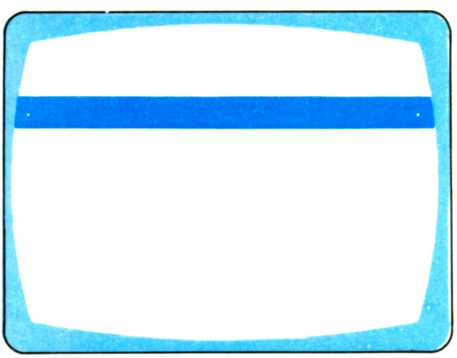
La mayoría de los micros modernos son capaces de presentar la información textual o gráfica en pantalla de muy diversas maneras.

Aparte de lo normal (letras negras sobre fondo blanco) se pueden invertir los colores de fondo y de primer plano.



**Pantallas de blanco y negro.** Tenemos aquí un ejemplo de escritura normal e invertida. La forma de conseguirlo varía con cada ordenador.

**Pantalla de color.** Dejar lista una pantalla de color puede resultar arduo si usamos un televisor corriente. Habrá que ajustarlo bien. Luego, ensayaremos las instrucciones de color, cambiando los del fondo y primer plano.



mado Forth (véase Módulo 32), y teclearle los anteriores programas no tendría ningún efecto: no entendería una palabra.

## Uso del color

Como ya estamos un poquito familiarizados con la programación y con la distribución de las te-

clas, podemos detenernos en algunas otras cosas que es capaz de hacer nuestro ordenador. Como vimos en el Módulo 7, para obtener la mejor imagen posible en el televisor es preciso sintonizar el receptor. Pero la mayor parte de los ordenadores actuales son capaces también de generar una amplia gama de colores. Los ordenadores suelen mostrar textos o números en la pantalla en lo que se llama «campo invertido», es decir caracteres blancos sobre fondo negro. En el Spectrum, sin embargo, se empieza con caracteres negros sobre fondo blanco.

Busquemos en el manual del usuario cómo se pone el ordenador en campo invertido (varía de una máquina a otra) y pulsemos la tecla SPACE (a veces, en vez de una tecla es una barra, como en las máquinas de escribir), que suele estar situada en la parte inferior del teclado. Comprobaremos que aparece en pantalla una línea ancha. Para cambiar el color de los caracteres hay que consultar de nuevo el manual del usuario. En el Commodore 64, por ejemplo, hay que pulsar la tecla CONTROL y una de las teclas numéricas. El 3, por ejemplo, produce caracteres en rojo, y otros números producen diferentes colores. Si seguimos pulsando la tecla SPACE obtendremos en pantalla una línea roja ancha. Repitamos el proceso para todos los colores posibles y terminaremos con una especie de arco iris en la pantalla. Conviene seguir experimentando hasta habernos familiarizado totalmente con las instrucciones que gobiernan el color en nuestra máquina particular.

Una vez en pantalla todos los colores, podemos ajustar el televisor para que salgan lo mejor posible. Tratemos de mover los mandos de contraste, brillo y densidad de color del receptor hasta dar con la combinación que sea más de nuestro agrado. Conseguido esto, no tendremos que volver a ajustarlo, a menos que nuestro televisor sea incapaz de mantener una imagen idéntica o coherente durante cierto tiempo.

**Uso del sonido**

Otra característica de la mayoría de los ordenadores domésticos de hoy en día es el sonido.

Hay dos métodos de generar sonido: usando un altavoz incorporado al ordenador, o aprovechando el del televisor. Como era de esperar, el sonido procedente del altavoz del televisor es de mucha mayor calidad, por ser (normalmente) más grande y complejo.

Muchos de los juegos de tipo *arcade* (los de esas máquinas electrónicas que ponen en bares y salones recreativos) para ordenadores domésticos utilizan profusamente los efectos sonoros, y ello puede acabar por resultar molesto cuando se ha jugado unas cuantas veces. Si el sonido procede del televisor, siempre podemos anularlo. El modo en que se genera el sonido refleja en cierta medida el modo en que podemos controlarlo. Para centrarnos en sólo dos de los ordenadores más populares, «beep» es la única instrucción de que dispone el Spectrum para hacer ruido, mientras que el Commodore 64 lleva prácticamente un sintetizador incorporado.

Pero sea cual sea nuestro ordenador, tratemos de generar unos cuantos sonidos para familiarizarnos tanto con la gama disponible como con las instrucciones necesarias para generarla.

**Uso de las posibilidades gráficas**

Otra característica común a todos los microordenadores es la posibilidad de mostrar en la pantalla caracteres gráficos de uno u otro tipo. La mayor parte de los ordenadores llevan su juego de caracteres en la máquina misma, lo que permite generarlo al completo sin más que pulsar unas pocas teclas. Unas máquinas tendrán más posibilidades que otras, pero todas son fácilmente accesibles desde el teclado.

Sería imposible escribir un programa que generase todos los caracteres de cualquier ordenador, pero hay un modo de hacerlo en cada máquina en concreto. Hay una instrucción (una palabra que

le dice al ordenador lo que tiene que hacer), llamada en la mayor parte de las máquinas CHR$, que hace que el ordenador muestre un carácter (como operación distinta de realizar una operación aritmética del tipo que sea). Cualquier carácter, como las letras de la A a la Z o los números del 0 al 9, y cualquiera de los caracteres gráficos especiales del teclado se pueden escribir con una sentencia CHR$. Por poner un ejemplo, la A mayúscula suele venir representada por el código 65, que equivale a una A, y por eso escribe tal carácter (véase Módulo 13). Por tanto, si tecleamos PRINT CHR$ (65) nuestro ordenador escribirá A. En este caso, lo más sencillo hubiera sido pulsar la letra sin más, pero mediante esta instrucción, poniendo entre paréntesis distintos números, empezaremos a descubrir las posibilidades gráficas de nuestro ordenador. Ensayémoslo a ver qué efectos se producen.

Igual que en los casos del color y del sonido, el único modo de aprender es experimentar con estas posibilidades gráficas, y conviene volver a subrayar que de ningún modo puede salir dañada la máquina por esto. El siguiente paso es tratar de escribir un programa sencillo utilizando estas posibilidades. Tratemos de incorporar el color, el sonido y los gráficos en todos nuestros programas hasta llegar a familiarizarnos con ellos, pues los programas saldrán ganando.

Si con un ordenador no se pudiera hacer más que lo hasta ahora descrito, realmente no tendría mucho sentido comprarse uno. Después de todo, cada vez que lo apaguemos se perderá todo lo que hayamos introducido. Al contrario, tenemos abiertas varias opciones para cerciorarnos de que lo tecleado se puede guardar y volver a recuperar cuantas veces haga falta. En el módulo siguiente veremos la forma de guardar programas y de conectar al ordenador los distintos periféricos que nos permiten volver a cargarlos.

# LOS MAGNETOFONOS

El medio más corriente de guardar lo que hemos introducido en el ordenador a través del teclado es el magnetófono doméstico ordinario. Es el más barato de todos y muchas personas lo tienen ya en casa antes de comprar el ordenador.

La única excepción real se plantea si somos propietarios de un Commodore. En este caso, tendremos que comprar un magnetófono especial fabricado por Commodore.

**La *interface* con el magnetófono**

Los propietarios de casi todas las demás marcas de ordenadores no tienen que afrontar el gasto de comprar un magnetófono especial, o por lo menos les basta con un modelo barato. Sin embargo, con los muy baratos puede haber algunos problemas, sobre todo si tienen control automático de tono. Algunos ordenadores son un tanto quisquillosos en lo que se refiere a estas cualidades sonoras, y se negarán a aceptar un programa si el tono no es el adecuado. Otro aspecto, que es importante tener presente si vamos a comprar un magnetófono específicamente para usarlo con nuestro ordenador, es la compatibilidad de enchufes y conectores. Conviene cerciorarse de que el magnetófono tiene las clavijas adecuadas para admitir los cables que han venido con nuestro ordenador.

Utilizando dichos cables, conectemos el magnetófono al ordenador como se explicó en el Módulo 7, e introduzcamos una cinta en él. Vale cualquier cinta, aunque habrá menos problemas si se usan de una marca de buena calidad. También se venden cintas especiales de corta longitud, que son ideales para almacenar multitud de programas cortos, ahorrándonos muchas pérdidas de tiempo en búsquedas y rebobinados. Téngase siempre presente que un programa sólo se puede cargar a la velocidad normal para oír música. Un programa largo, como suelen serlo, por ejemplo, los de juegos, puede tardar en cargarse un tiempo desesperantemente largo.

**Comprobación de la *interface***

Una vez conectado el magnetófono al ordenador, con una cinta montada y completamente rebobinada, tendremos que introducir un pequeño programita para comprobar que todo funciona satisfactoriamente. O lo escribimos nosotros mismos o usamos el siguiente ejemplo:

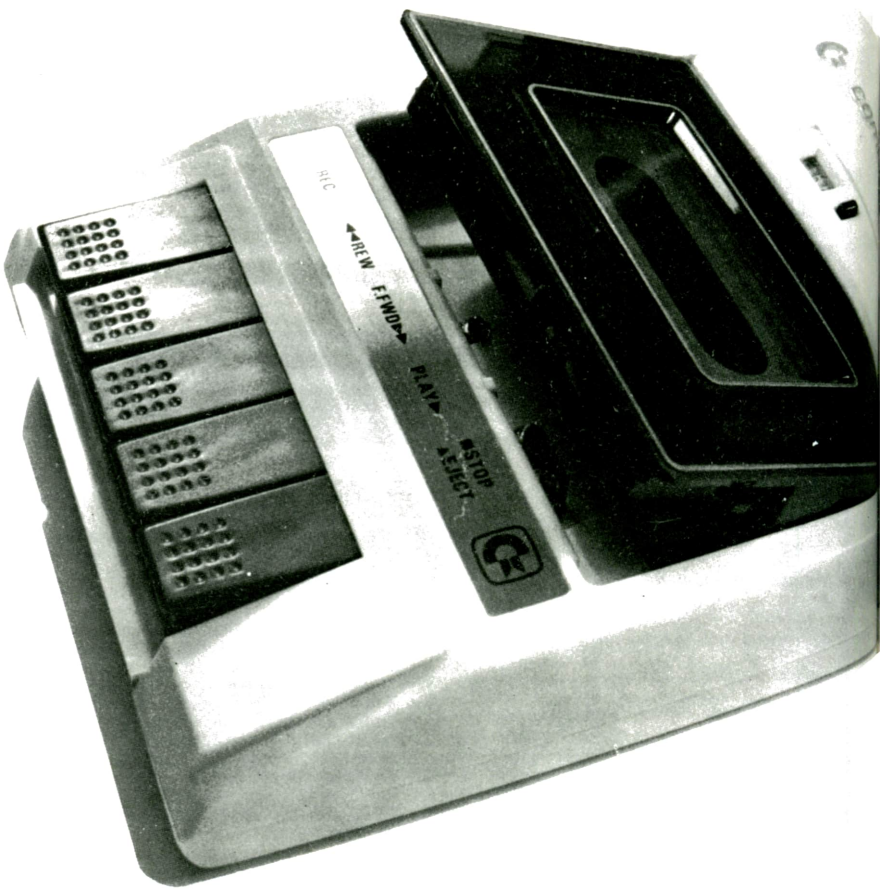
```
10 PRINT "ESTO ES UNA PRUEBA"
20 PRINT "DE MI CASSETTE"
30 GOTO 10
```

Si, una vez tecleado, pedimos la ejecución del programa mediante la instrucción RUN, lo único que sucede es que las dos frases ESTA ES UNA PRUEBA y DE MI CASSETTE aparecen sin cesar en la pantalla, porque hemos creado lo que se llama un «bucle» en el programa (véase Módulo 27). Para evitar que el mensaje siga apareciendo habrá que detener la ejecución del programa pulsando la tecla BREAK. Algunos ordenadores usan una tecla STOP para este propósito, y aún otros una tecla ESCAPE. Todas ellas, en cualquier caso, sirven para lo mismo.

Para guardar en cinta este programa, lo prime-

ro es decidir cómo lo vamos a llamar, por ejemplo TEST PRG. A continuación tecleamos SAVE "TEST PRG" y pulsamos la tecla ENTER o RETURN. El ordenador nos dirá qué hay que hacer después, que será apretar los mandos PLAY y RECORD del magnetófono, igual que si fuéramos a grabar música.

Si el magnetófono tiene contador, no olvidemos apuntar el número que marca cuando grabamos un programa. Esto puede ahorrarnos mucho tiem-




**El magnetófono de Commodore.** Con la mayoría de los micros domésticos se puede utilizar cualquier magnetófono de calidad aceptable, aunque tal vez sea preciso colocar el mando del volumen en una posición

muy concreta. Hay micros que sólo funcionan con los magnetófonos que sus fabricantes suministran; suelen ser caros, pero más fiables. El aparato que aparece en la figura se usa con los ordenadores Commodore.

po al ir a buscar un programa concreto. Las cintas funcionan con el ordenador igual que las de música: es imposible encontrar el principio de un programa (o pista de grabación) si no se sabe el número del contador. Si no hemos anotado la posición exacta de los programas, no todo está perdido. Como cada uno tiene su nombre, que debe ir entre comillas, el ordenador puede ir recorriendo todas las cintas hasta dar con el nombre del programa buscado. El procedimiento es muy lento, no obstante, porque no se puede usar el mando de avance rápido.

Volviendo al programita de antes, transcurridos unos pocos segundos la cinta se detendrá, y el ordenador nos indicará que todo ha ido bien. No nos dejemos engatusar: puede no ser verdad. La única forma de saber si el programa está correctamente guardado es borrarlo de la memoria y cargarlo de nuevo. En la mayor parte de las máquinas el borrador se consigue mediante la instrucción NEW. Hecho esto, comprobaremos si el programa se ha grabado realmente tecleando LOAD "TEST PRG", el nombre que asignamos al programa en el momento de guardarlo. En otras máquinas esta orden de recuperación puede adoptar formas distintas. No olvidemos rebobinar antes la cinta hasta más allá del punto de comienzo del programa que queremos cargar.

Sigamos de nuevo las instrucciones del ordenador, que nos dirá que pulsemos el mando PLAY del magnetófono, y esperemos luego hasta que nos indique que ha terminado de cargar. Entonces podemos teclear la instrucción LIST, que ordena a la máquina sacar un listado del programa por la pantalla, para comprobar si está completo. Si todo ha ido bien, podemos afirmar que nuestra *interface* de magnetófono funciona correctamente y pasar a guardar programas mayores.

Por el contrario, si el ordenador no hace nada después de haber pulsado el mando PLAY, paremos la cinta, rebobinemos, e intentémoslo de nuevo. Esta vez, sin embargo, subiendo el volumen un

poco y ensayando otra posición en el mando de tono. En unos cuantos intentos terminaremos por dar con la combinación adecuada, y así guardar y recuperar programas sin más problemas.

Como muchos fabricantes y tiendas venden cintas para ordenador pregrabadas, nos interesará que el magnetófono funcione lo mejor posible con nuestro ordenador. El tiempo que dediquemos a ajustarlo bien será una buena inversión. Existen, sin embargo, maneras de almacenar y recuperar programas más eficientes que el magnetófono, como comprobaremos en el siguiente módulo.

# DISCOS Y CARTUCHOS

Las unidades de disco no son todavía tan comunes en los ordenadores domésticos como los magnetófonos a causa de su precio, pero lo cierto es que resuelven todas las limitaciones que tiene el almacenamiento de la información en *cassette.* En primer lugar, en velocidad de operación y en capacidad de almacenamiento son muy superiores a las *cassettes.* Cuentan además con la ventaja suplementaria de que no es necesario recordar la posición de todos los programas: el ordenador lleva la cuenta. El aumento en velocidad y eficiencia se debe a que el disco gira a unas 300 revoluciones por minuto y la información no se almacena secuencialmente, sino aleatoriamente. El uso de la palabra «aleatorio» puede causar una impresión errónea. Lo único que significa en este contexto es que la cabeza de «lectura-escritura» que lleva cada unidad de disco (véase Módulo 22) puede desplazarse sobre la superficie del disco, bajo control del ordenador, y acceder a la información situada en cualquier parte. Estas propiedades han convertido al disco en elemento indispensable de todo sistema informático de gestión, y paulatinamente se va introduciendo también en el mercado del ordenador doméstico.

Pese a todas las ventajas que ofrece el almacenamiento en disco, su manejo es muy sencillo y se utilizan las mismas instrucciones que en el

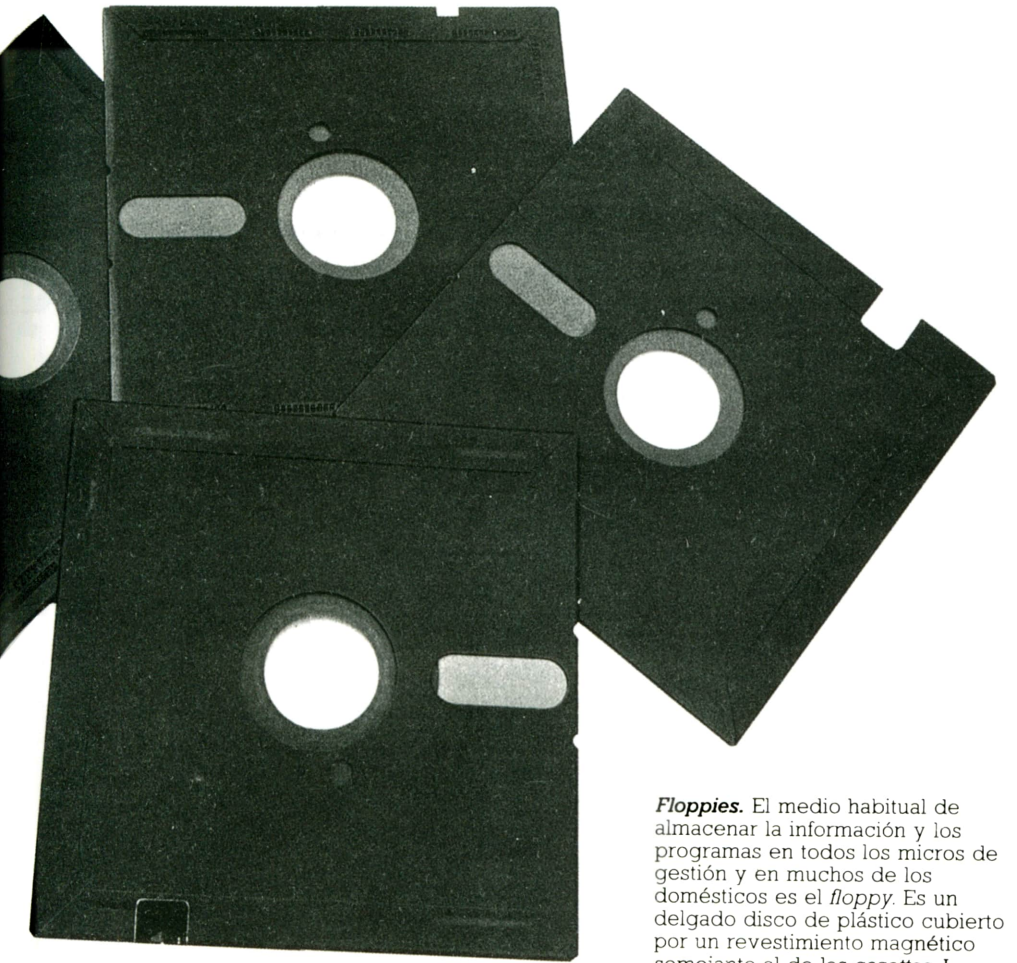
magnetófono (véase Módulo 9) para cargar y guardar programas, con una importante adición. Todo lo que se conecta a un ordenador debe tener un número o nombre de dispositivo para que el ordenador pueda distinguirlo de los demás. Por ello, para indicarle que deseamos guardar un programa en disco y no en cinta, hay que teclear una instrucción tal como SAVE «B:TEST PRG». En este ejemplo, B es el nombre de dispositivo de uno de los discos conectados al ordenador, y TEST PRG el nombre que hemos dado a ese programa concreto. El formato y la redacción de estas instrucciones «save» difiere de una máquina a otra, pero son siempre muy sencillos.

**Los cartuchos**

Una alternativa al disco es el cartucho (véase Módulo 21). Los cartuchos se introducen en una ranura de la parte posterior de los ordenadores y, una vez instalados, «se hacen cargo» verdaderamente de la máquina. Los cartuchos se utilizan casi siempre para cargar programas en el ordenador más que para guardar programas escritos por nosotros mismos, aunque en algunas máquinas esto también es factible. Los programas que vienen en los cartuchos suelen ser más complicados que los escritos por el aficionado corriente, y los costes de fabricar y embalar el cartucho se combinan para hacer de éste un dispositivo caro. Los fabricantes de *software* prefieren sin embargo este sistema porque, una vez el programa dentro del cartucho, no es posible copiarlo.

**Publicaciones de informática**

Constituyen una cantera importantísima de programas y hay que tenerlas muy presentes. Existen ya tantas revistas especializadas en informática que es tarea fácil seleccionar las que contienen los mejores listados de programas para nuestra máquina. Tal ha sido la proliferación de este tipo de revistas, que es muy probable que encontremos alguna dedicada exclusivamente al orde-



***Floppies.*** El medio habitual de almacenar la información y los programas en todos los micros de gestión y en muchos de los domésticos es el *floppy*. Es un delgado disco de plástico cubierto por un revestimiento magnético semejante al de las *casettes*. La información se graba en este revestimiento y se lee mediante una cabeza lectora-escritora.

nador que poseemos, sobre todo si es de los más populares. Una vez reunidos, comprobaremos que tenemos listados más que suficientes para teclear.

Al teclear un programa es inevitable cometer errores que habrá que corregir posteriormente (un proceso llamado «depuración»).

Tras varias horas de teclear listados, y antes de que llegue alguien y tire del cable, echando a perder nuestro trabajo, hay que transferir el programa a *cassette* o a disco (véanse Módulos 21 y 22). Conviene siempre guardar el programa antes de intentar ejecutarlo, por si alguno de los errores cometidos (o que ya contuviera el programa) provoca una «caída» del sistema. Se dice que se produce una caída cuando el programa contiene algo que ocasiona una pérdida irreparable de información.

Cuando hemos terminado de teclear el programa y guardado una copia, si estamos convencidos de que no contiene errores que vayan a dar problemas, la diversión comienza.

El programa que tenemos delante puede ser de un juego o de una aplicación más seria, tal como la contabilidad doméstica. Tanto las revistas como los libros, sin embargo, atienden también a los usuarios comerciales de microordenadores, por lo que a veces traen programas de aplicación sobre asuntos tales como las bases de datos (véase Módulo 34) y las hojas de cálculo (véase Módulo 35). Pero sea cual sea el programa, una vez en la máquina (con una copia bien guardada) podemos comenzar a modificarlo a nuestro antojo. La mayor parte de ellos se publican dando por sentado que van a ser personalizados por el usuario. Y cuando ya nos sintamos satisfechos, podemos guardarlo junto con el original, obteniendo así dos programas por cada uno publicado.

Tratemos de modificar cuantos programas sea posible, aprendiendo del original a medida que lo vamos haciendo. Ninguna otra cosa nos enseñará más sobre los secretos de la programación. Y no hay motivo para limitarnos a programas escritos especialmente para nuestra máquina. La mayoría de los microordenadores domésticos utilizan el mismo o parecido dialecto del Basic, así que estudiemos los programas escritos para otras máquinas, y con tiempo y paciencia conseguiremos adaptarlos a la nuestra.

# CONSEJOS SOBRE PROGRAMACION

Una vez adquirida cierta confianza y familiarizados plenamente con nuestro ordenador, su teclado y las instrucciones necesarias para gobernar sus variadas funciones, escribir programas es cuestión de adoptar la actitud adecuada.

**Planear el programa**

Cuando por fin nos ponemos a escribir programas para nuestro propio uso, lo primero es determinar la esfera de la programación en que vamos a concentrar nuestros esfuerzos durante una sesión concreta. No se puede, por ejemplo, empezar con la idea de escribir un programa de juegos del tipo que sea (véase Módulo 29) y luego, a mitad de camino, pensar de repente que nuestro tiempo estaría mejor empleado si lo dedicásemos a escribir un «paquete» de cálculo integrado (véase Módulo 35). Así que elijamos una especialidad y propongámonos dedicar las siguientes horas, o el tiempo que podamos estar ante el teclado, a programar eso y no otra cosa.

**Los algoritmos**

Es posible que a algunas personas les baste sentarse al teclado y empezar con su programa, modificándolo a medida que lo escriben. Crearán probablemente programas razonables, que fun-

cionan, pero que, sin duda, podrían estar mejor estructurados, ser más rápidos y hacer más cosas.

Para redactar un programa eficiente es preciso planear exactamente lo que queremos que haga, sin desanimarnos porque alguna parte se niegue obstinadamente a funcionar como debiera. Sin duda es que estamos haciendo algo mal o atacando el problema erróneamente o desde un ángulo incorrecto.

Planeando de antemano lo que queremos conseguir se atenúa el riesgo de que surjan problemas de este tipo. La forma más usual de planificar un programa es trazar un organigrama de sus diversas partes y del modo en que se interrelacionan —a esto es a lo que se llama algoritmo (véase Módulo 26). Si pensamos escribir un programa de juegos no bastará con poner «habrá cuarenta invasores y una nave espacial». Esto no dice nada sobre el programa ni sobre el modo de desarrollarlo.

Todas las secciones de que conste el programa deben estar perfectamente estudiadas antes de sentarse al ordenador. Volviendo a este mismo ejemplo de los juegos, necesitaremos una «rutina» que mueva la nave de izquierda a derecha en la parte inferior de la pantalla, y tenga en cuenta el hecho de que no queremos que se salga de la pantalla ni por un lado ni por otro. Hay que pensar cómo vamos a escribir esta «rutina», para mover la nave y conocer la distancia que la separa de los bordes. Si nos lanzamos a teclear sin más, lo probable es que el programa no funcione, y aunque por casualidad llegara a hacerlo, no nos habría enseñado nada sobre programación.

**Cómo organizar un programa**

Cuando nos hayamos acostumbrado a escribirlo todo de antemano, nos daremos cuenta de que este proceso implica mucho más de lo que en un principio se adivina. Según vayamos adquiriendo experiencia, empezaremos a constatar que ciertas partes de nuestros programas son similares a

otros programas que ya hemos escrito. Por tanto, en lugar de repetir el mismo trabajo una y otra vez, podemos echar mano de «rutinas» ya utilizadas y comprobadas, quizá con alguna ligera modificación. Con esto aprenderemos otra lección sobre el arte de escribir programas: cerciorémonos siempre de que cuando vayamos a usar un programa escrito tiempo atrás seremos capaces de saber exactamente lo que hace cada una de sus partes.

Para facilitarlo, los ordenadores domésticos cuentan con una sentencia llamada REMARK (normalmente abreviada REM ), que permite incluir comentarios en un programa. Por ejemplo, al principio de la «rutina» que «mueve la nave a la izquierda» podríamos insertar un par de sentencias como estas:

```
1000 REM ESTA PARTE DEL PROGRAMA
2000 REM MUEVE LA NAVE A LA IZQUIERDA
```

De este modo siempre sabremos lo que hace esa parte del programa. Incluyendo estos comentarios en los listados, podremos volver a orientarnos en seguida y localizar las «rutinas» útiles que pudieran adaptarse a unas circunstancias análogas en otro programa.

El hábito de poner comentarios en los programas no sólo nos ayudará a nosotros, sino a cualquiera que vaya a usarlos.

**Las subrutinas**

Un «truco» de programación muy útil es guardar todas las «rutinas» comunes que vemos repetirse una y otra vez. Utilicemos un disco o una *cassette* aparte (dependiendo de cual sea nuestro medio de almacenamiento, véanse Módulos 9 y 10), y en poco tiempo contaremos con una colección de rutinas o, como es más correcto llamarlas, «subrutinas», de gran interés. Una «subrutina» es, en esencia, 'un pequeño programa autónomo que cumple

una misión específica'. Por ejemplo, una rutina para reducir números al formato de dos decimales. Tenemos un valor numérico, pongamos A, igual a 12.7684239, que queremos que tenga un aspecto más sencillo al aparecer en pantalla. Nuestra subrutina, pues, hará A igual a 12.77.

En un programa largo es posible que nos interese utilizar varias veces esta rutina, y sería un engorro tener que volver a teclearla cada vez. Para esto existe la subrutina, un programa corto que lleva a cabo una función útil, y que lo hace muchas veces dentro de un programa grande. La sentencia de Basic que se utiliza para llamar a una de estas subrutinas es la GOSUB, que transfiere el control del programa al pequeño subconjunto para realizar la función deseada. Para volver al flujo principal del programa se utiliza RETURN.

Construyendo un conjunto de rutinas como éste, el tiempo invertido en escribir programas se puede reducir considerablemente, y planeándolo todo de antemano también se abreviará mucho el tiempo de desarrollo. Pero, ¿de dónde proceden todas estas rutinas? Obviamente, casi todas las habremos escrito nosotros mismos, pero son numerosas también las publicaciones que detallan programas de tipo muy diverso. Hay revistas y libros que nos ayudarán a escribir programas, y asimismo se puede conseguir *software* a través de los grupos de usuarios.

Cuando haya pasado el tiempo y nuestro trabajo de programación dé unos resultados cada vez mejores, acabaremos por escribir rutinas que se podrán usar en muy diversas esferas de aplicación del *software*. Entre ellas, colección de rutinas financieras, rutinas de movimientos en la pantalla, etc., que podremos incluir en nuestros programas a medida que las necesitemos.

**Sistema de archivar programas**

Como dijimos antes, lo ideal es que todas estas rutinas estén guardadas en *cassette* o en disco para usarlas cuando convenga. En cuanto comencemos

a elaborar (o simplemente comencemos a encontrarnos) estos programas, nos convenceremos de la necesidad de organizar un sistema de archivo que permita encontrar el programa adecuado en el momento oportuno.

Un dispositivo ideal para estos fines es una impresora (véase Módulo 23). Las impresoras pueden ser caras, llegar incluso a costar más que el mismo ordenador, pero contribuirán sensiblemente a organizar nuestra colección de subrutinas (aparte de que su campo de aplicación es mucho más amplio). Tomemos, por ejemplo, un conjunto de programas utilizables en programación financiera, guardados todos en el mismo disco o cinta. Si tenemos disco, sería interesante conservar junto a él una lista en papel de los programas que contiene. O, si usamos *cassettes,* la lista de los programas que contiene con las posiciones en que comienzan. Si no ponemos una impresora a nuestro ordenador, todas estas listas de programas tendremos que hacerlas a mano, lo que puede resultar muy laborioso.

**Normalización de los programas**

Terminará por llegar el momento en que la rutina que precisamos para realizar una función determinada no case con otro programa si no es a costa de drásticas modificaciones. Esto ocurrirá posiblemente varias veces en el curso del desarrollo de un programa, pero hay formas de aminorar el riesgo de que suceda. El primero y más importante es la normalización. Todos los ordenadores domésticos tienen sus peculiaridades al programar en Basic. Así, no es posible presentar un conjunto de reglas rígidas, estrictas, que se puedan aplicar invariablemente a todos los ordenadores. Pero sí que pueden formularse ciertas generalizaciones que son lo suficientemente importantes como para resultar útiles y quizá esenciales.

Tomemos por ejemplo los nombres de las variables. No son más que nombres que damos a un número, a una cadena de caracteres, etc., y que

permiten al ordenador saber de qué conjunto de caracteres en particular estamos hablando. Supongamos que tenemos una sentencia como esta:

```
10 A$="CUALQUIER CADENA ANTERIOR"
```

Cuando se quiera imprimir CUALQUIER CADENA ANTERIOR basta poner A$. Se puede ir más lejos. A$ significará siempre lo mismo para el ordenador, pero a los seis meses ya se nos habrá olvidado. Hubiéramos podido poner:

```
10 CCA$="CUALQUIER CADENA ANTERIOR"
```

Con lo que las letras del nombres de la variable nos proporcionan las iniciales de las palabras de la cadena. Imaginemos una rutina para sumar dos números y escribir el resultado:

```
1000 A=5:B=10
1010 ADD=A+B
1020 PRINT ADD
```

Se ha usado la variable ADD para sumar los dos números. Aparte de indicarle al ordenador lo que tiene que hacer, nos dice de lo que se trata.

Tropezamos aquí, sin embargo, con un problema: la mayoría de los ordenadores no reconocen más que las primeras letras del nombre de una variable. Aunque maneje sin protestas algo llamado ADD, en memoria (Módulo 19) guardará solamente AD. No importa demasiado mientras no tengamos otra variable que se llame, por ej. ADDA.

Otra acción potencialmente peligrosa a la que hay que estar atentos es la de incluir instrucciones del Basic en los nombres de nuestros programas. No se pueden usar, por ejemplo, ni IFFY, ni PRINTER, ni CATALOG, porque contienen respectivamente las sentencias del Basic IF, PRINT y LOG. Si utilizamos en nuestras subrutinas nombres

## CONVIENE RECORDAR

**Algunas breves definiciones de términos que figuran en esta sección.**

**Basic.** Lenguaje de programación más difundido. Lo llevan casi todos los micros. Muy fácil de aprender para los principiantes.
**Magnetófonos.** Muchos micros domésticos utilizan *cassettes* para almacenar programas en cintas de audio.
***Hardware.*** Los componentes físicos del sistema.
**Teclado.** Parecido al de las máquinas de escribir. Sirve para comunicarse con el ordenador.
**Modulador.** Para convertir las señales de vídeo del ordenador en señales sintonizables por el televisor es necesario un modulador.
**Monitor.** Pantalla especial utilizable en lugar del televisor. Más caros, pero con calidad de imagen muy superior.
***Software .*** Las instrucciones o el programa que indican al *hardware* del ordenador lo que tiene que hacer.

de variables válidos será más sencillo formar con todas ellas un programa coherente y fácil de comprender.

¿Qué tipo de subrutinas se debe usar? Muy diversas son las aplicaciones, y muchos los problemas que hay que resolver, pero comprobaremos que algunos se presentan repetidamente. Para introducir información en el ordenador, por ejemplo, hay multitud de rutinas publicadas, y es una de las primeras cosas que debemos tener.

También son muy comunes las subrutinas de clasificación, y se puede decir que su variedad es tanta como la de los ordenadores. Cada una de ellas supera a las demás en una situación concreta, y por tanto conviene cerciorarse de que estamos utilizando la adecuada para nuestro problema particular. Si tarda mucho en clasificar, lo más probable es que estemos usando una inadecuada.

No temamos nunca modificar un programa que hayamos encontrado en una revista o nos hayan

dado. Y si vemos una rutina que es ideal para un problema que tenemos en estudio, pero escrito para una máquina distinta, merece la pena tratar de adaptarlo a la nuestra. Las diferencias existentes entre muchos dialectos del Basic son ligerísimas; si tropezamos con la instrucción CLS y buscamos en vano por nuestro teclado una tecla que tenga las mismas letras, pensando un poco y observando cuidadosamente el listado en seguida caeremos en la cuenta de que CLS significa CLEAR SCREEN (borrar pantalla). Lo único que hay que hacer es poner en su lugar el código que tenga nuestro ordenador para esta instrucción. Con sólo pensar un poco daremos con la forma de corregirlo todo. Y así es como se debe proceder para escribir programas: concretar el área de aplicación, decirse a uno mismo que se va a escribir eso y no otra cosa, y atenerse a unas cuantas reglas sencillas.

# DETECCION DE AVERIAS

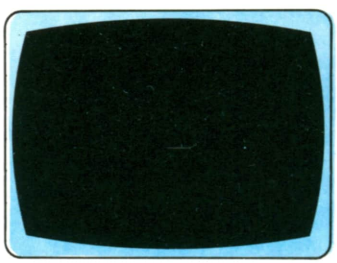
Teniendo en cuenta la complejidad electrónica de los modernos micros, las averías completas del sistema son sorprendentemente escasas. Si, a pesar de todo, no estamos de suerte, las orientaciones que ofrecemos en el recuadro de las páginas 114-115 nos ayudará a resolver rápidamente el problema. Si algunos de los consejos parecen obvios, es porque normalmente es lo obvio lo que falla. Las orientaciones cubren la mayor parte de las averías que pueden presentarse cuando se arranca el ordenador para iniciar una sesión. Si, después de comprobar todo esto, algo sigue yendo mal, hay que cerciorarse de que el televisor o monitor esté enchufado, o de que el cable de vídeo siga en su sitio, o de que ningún fusible haya saltado. Si sigue negándose a funcionar, dirijamos nuestra atención a los periféricos, la *cassette* o el disco, por ejemplo. ¿Están bien conectados y en las clavijas adecuadas? Si el ordenador aún no responde, probablemente haya que recurrir a un profesional.

**Caída del sistema**

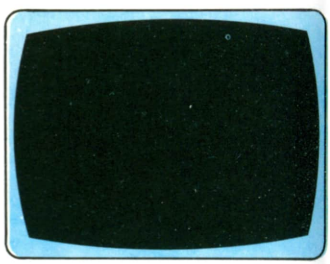
Suponiendo que todo haya funcionado bien durante algún tiempo, y un día el sistema «caiga» (palabra que significa que algo grave le ha sucedido al ordenador o a alguno de sus componentes) habrá que adoptar otro conjunto de medidas.

Lo primero, averiguar si el fallo es nuestro, del

Si nuestro micro no responde como debería al comenzar una sesión, lo probable es que algo esté estropeado. Los micros modernos son extremadamente fiables, y por ello es preferible buscar el problema en lo más obvio: hilos sueltos, cables

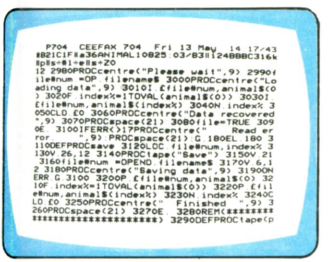


**Síntoma.** Pantalla sin imagen.
**Causa.** Pueden ser varias. El ordenador no está encendido, o no está conectado a la fuente de alimentación, o tiene algo estropeado.
**Remedio.** Cerciorarse de que el ordenador está encendido, y de que está conectado a la clavija. Comprobar el fusible. Suele estar situado en la parte posterior y se accede a él con destornillador. Si el fusible ha saltado, hay que cerciorarse de usar el adecuado. Caso de no estar seguros, es preferible consultar al vendedor.

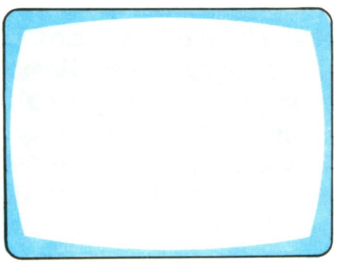


**Síntoma.** Pantalla sin imagen, aunque el ordenador esté encendido y funcione correctamente.
**Causa.** Pueden ser varias. El televisor no está bien sintonizado, los cables no están conectados, el modulador no se ha enchufado, o el cable de vídeo no está en su sitio. Son causas fáciles de corregir: comprobemos las conexiones.
**Remedios.** Apagar el micro unos segundos y encenderlo de nuevo. Con esto se corrigen a veces algunos problemas internos. Si no, comprobar que el televisor está bien sintonizado, los cables están bien conectados, y que también lo está el cable de vídeo. Si no tenemos los cables en condiciones, lo más probable es que este tipo de avería acabe convirtiéndose en una plaga.

desconectados, fusibles fundidos. Los problemas de sonido y color derivan normalmente del televisor, no del ordenador. Sólo una vez comprobadas todas las causas externas y el *software* consulte a su distribuidor.




**Síntoma.** Excesivo ruido de fondo del TV o falta de sonido.
**Causa.** Mal ajuste del mando de volumen del televisor.
**Remedio.** Fácil: reajustar el mando de volumen del televisor.



**Síntoma.** Imagen sin color o con color defectuoso.
**Causa.** El televisor no está bien sintonizado, o es preciso ajustar los mandos del color.
**Remedio.** Si el ordenador sólo trabaja en blanco y negro, poca cosa puede hacerse. Sin embargo, si es capaz de dar imágenes en color, se trata simplemente de volver a sintonizar el televisor y probar todos los colores con el método recomendado en el Módulo 7 (escribir una serie de espacios en cada uno de los colores disponibles en nuestro ordenador).

ordenador o de alguno de los periféricos que tiene conectados. Si hemos estado haciendo experimentos en lenguaje de máquina, o metiendo valores aleatoriamente en distintas posiciones de la memoria con la instrucción POKE (véase Módulo 27), lo más probable es que los causantes hayamos sido nosotros. Ciertas zonas de la memoria del ordenador son muy sensibles a los valores que se ponen en ellas. Si hemos estado programando en lenguaje de máquina, las posibles causas de la caída son múltiples. En todos los programas escritos en este lenguaje hay que atenerse a ciertas reglas. La principal es guardar siempre el programa antes de tratar de ejecutarlo. Esto es aplicable también a los programas en Basic.

Una de las causas más corrientes de caída originada por un programa en código de máquina es haber entrado en un bucle sin fin, por haber olvidado devolver el control mediante un RETURN al programa principal.

**Problemas con el Basic**

Si lo que ha originado la caída no es el código de máquina, quizá provenga de un programa en Basic. Hay que seguir cierto número de reglas si se quiere localizar el error. Los mensajes de error generados por el propio ordenador pueden darnos alguna pista. El más común, aunque la redacción exacta puede variar de una máquina a otra, es SYNTAX ERROR IN XXX, donde XXX es el número de la sentencia donde se produjo el error. Hay otros mensajes de error, y casi siempre apuntan a la posible solución. Si, por ejemplo, nos dice OUT OF DATA ERROR IN XXX, es que el ordenador espera encontrar algún dato en la línea XXX que no está ahí, y que por tanto debemos poner. O quizá saque un mensaje NEXT WITHOUT FOR IN XXX. Esto significará que hemos indicado al ordenador que dé el siguiente paso de un bucle FOR...NEXT que no hemos siquiera arrancado.

Si no se produce ningún mensaje de error, y el programa no funciona, una buena solución es in-

cluir sentencias STOP (o BREAK) en diferentes puntos del programa. Si acertamos a poner una de ellas inmediatamente detrás de la parte del programa que no funciona, la ejecución del mismo se detendrá y podremos examinar las variables que estemos usando, las sentencias ya ejecutadas, cambiar lo que sea e intentarlo de nuevo. Empleando reiteradamente este método acabaremos por resolver casi todos nuestros problemas.

**Problemas con la máquina**

Si no encontramos ningún error de programación obvio ni en Basic ni en código de máquina, ¿qué otra cosa puede ir mal? Si estamos convencidos de que nuestro programa funciona bien, o ya nos ha funcionado antes correctamente, será razonable suponer que es la máquina, o alguno de sus componentes, la culpable. Lo más probable es que se trate de algún cable suelto o algo por el estilo. Si es así, lo más probable es que hayamos perdido el programa en el que estábamos trabajando o que se estaba ejecutando en ese momento: si hemos tomado la precaución de hacer una copia de seguridad, nada se habrá perdido.

Si todos los cables están bien, es posible que alguno de los conectores de tales cables se haya astillado ligeramente. Llevaría algún tiempo un poco suelto y un último golpe lo ha partido. Este tipo de avería puede resultar difícil de localizar, pero si comprobamos que un cable es el origen del problema, la única solución es comprar otro y rehacer la conexión. Si todos los cables están bien y en su sitio, el siguiente paso es comprobar los fusibles, que pueden saltar en cualquier momento sin ninguna manifestación externa salvo que todo se detiene. Siempre conviene, como precaución, tener en casa fusibles de repuesto.

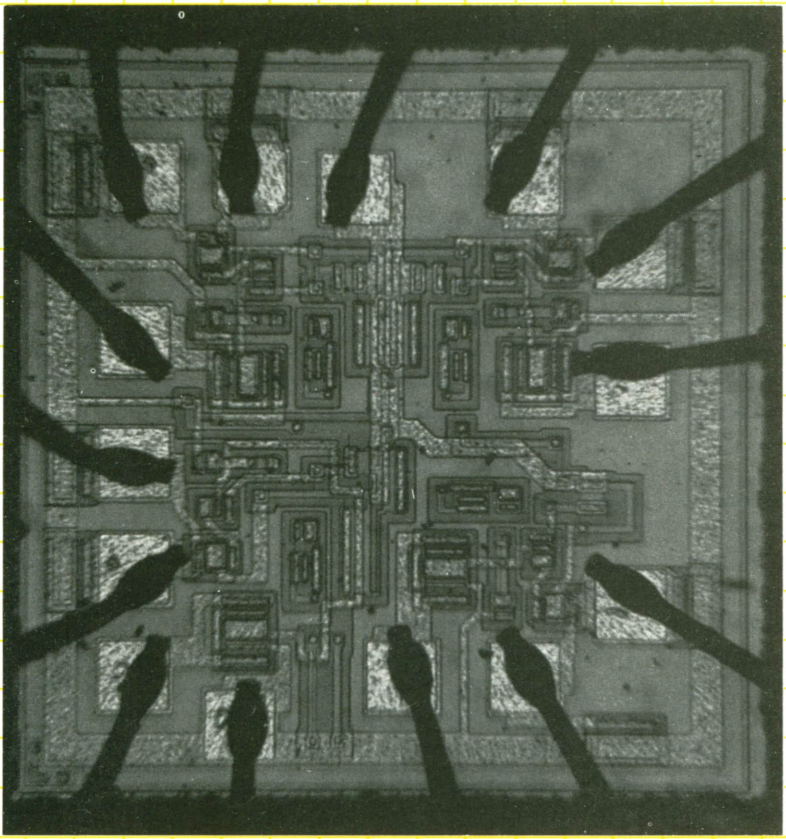
Si el problema no es de cables ni de fusibles, ¿no será que estamos intentando hacer algo imposible? Tratando, por ejemplo, de guardar un programa en disco, pero usando un código de dispositivo que no está en uso. Esto es muy fácil que

suceda, basta equivocarse de tecla. Los errores de este tipo se producen sobre todo cuando hemos estado tecleando el mismo código tantas veces que esa acción se ha convertido en algo automático. Si esto sucede, el ordenador tratará de obedecer una orden que no puede cumplir.

Una vez examinadas todas las causas, probables y no tan probables, mencionadas aquí, si el sistema sigue negándose a funcionar, repasémoslas todas una vez más y después solicitemos la ayuda de un profesional. Si está averiado cualquiera de los componentes internos del ordenador o de algún periférico, no es recomendable que levantemos la tapa para investigar. Podemos hacerlo si nos empeñamos, pero si algo más se estropea la culpa será nuestra, y la mayoría de los comerciantes se negarán, con toda razón, a examinar una máquina manipulada.

En resumen, comprobemos en primer lugar siempre lo más obvio: cables que se aflojan, enchufes que se salen de las clavijas. Una sentencia que estamos seguros es correcta puede contener un pequeño error mecanográfico cuyos efectos se propagan por el resto del programa de forma catastrófica. Adoptando siempre las debidas precauciones y buscando el posible error con calma, reflexionando, pronto conseguiremos que en nuestro ordenador todo vaya sobre ruedas.

# COMO FUNCIONAN



# COMO «PIENSAN» LOS ORDENADORES

Todos conocemos la imagen que da la ciencia-ficción de los ordenadores: son capaces de hablar, de manifestar las cualidades humanas de simpatía, astucia, rencor, avaricia y hasta megalomanía. En la realidad, sin embargo, las cosas son bien distintas. No se puede decir que los ordenadores «piensen» en el sentido humano de la palabra, aunque expresiones como la de «inteligencia artificial» (véase Módulo 37) sugieren lo contrario. Es posible lograr que la conducta de un ordenador imite alguna forma limitada de pensamiento. Pero al fin y al cabo no es más que una máquina, aunque compleja, ingeniosa, rápida y potente.

**Los sistemas de numeración en informática**

Para resolver problemas, los ordenadores han de manejar números y, como son dispositivos electrónicos, de algún modo han de representarlos eléctricamente. Un modo de hacerlo es usar voltajes distintos para representar números diferentes: 1 voltio para el 1, 2 voltios para el 2 y así sucesivamente. Sin embargo, este método plantea ciertos problemas prácticos, sobre todo si van a manejarse números como el 100.000. Otro problema sería el dar los voltajes exactos para distinguir, pongamos, entre 10.001 y 10.002 y al tiempo entre 0,00002 y 0,00003.

Un sistema más sencillo sería representar el 1 por la presencia de una corriente eléctrica y el 0 por su ausencia. La desventaja es que así sólo podemos representar dos valores, 0 y 1, y por tanto se precisa un sistema de numeración totalmente distinto.

Estamos habituados a usar un sistema de numeración basado en el 10, el sistema decimal o denario. Tomemos por ejemplo el número 3.057. Usamos diez símbolos —del 0 al 9— y los colocamos en unas posiciones que les añaden un significado, las columnas de las unidades, decenas, centenas y millares (véase el recuadro de la página 123). Decimos que el sistema decimal es un sistema de numeración de «base 10» porque todo gira en torno al número 10.

Es posible construir un sistema de numeración que utilice cualquier otra base. Supongamos que queremos construir un sistema de numeración basado en el 16. El primer problema es la necesidad de contar con símbolos que representen los valores decimales 10, 11, 12, 13, 14 y 15. Podemos usar letras: A=10, B=11, C=12, D=13, E=14 y F=15. Contamos como en decimal: 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, hasta llegar a la F en la columna de las unidades. Sumando 1 nos da 16, así que pondremos un 1 en la columna de la izquierda y volveremos al 0 la de las unidades. Así, «10» en este sistema representa el número decimal «16» y «2A» es el decimal «42». Al sistema de numeración de base 16 se le llama «hexadecimal», y es uno de los utilizados en informática.

**El sistema binario**

Para usar sólo dos símbolos —0 y 1— necesitamos un sistema de numeración basado en el 2, el sistema «binario». Se cuenta, lo mismo que antes, incrementando las unidades hasta llevarnos una, momento en el que la columna de las unidades vuelve a 0 y la columna de la izquierda se incrementa en 1. Así pues, los primeros números en binario serían (entre paréntesis cada equivalente

decimal): 0 (0), 1 (1), 10 (2), 11 (3), 100 (4), 101 (5), etc. (véanse el recuadro de las páginas 124-125).

La notación binaria es la adecuada para hacer cuentas en el mundo de la electrónica. Para aclarar más el funcionamiento del binario, imagine una serie de lámparas (véanse las páginas 124-125) en la que las encendidas representan el 1 y las apagadas el 0.

De poco valdría, en cualquier caso, un sistema de numeración si no pudieran hacerse con él operaciones aritméticas. Pero la aritmética binaria funciona igual que su equivalente decimal, como puede verse:

| | |
|---|---|
| 1001 | 9 |
| + 101 | + 5 |
| 1110 | 14 |

**Bits y octetos**

A los dígitos se les llama en informática bits (del inglés *binary digit*). Un bit solo no representa gran cosa, por lo que se suelen agrupar en conjuntos de tamaño conveniente. En casi todos los microordenadores esta unidad es el «octeto» *(byte),* un grupo de ocho bits. La capacidad de la memoria del microordenador se mide por el número de octetos que puede almacenar, contando grupos binarios de 1.024 octetos, llamados kilooctetos, k-octetos o simplemente k. El tamaño máximo de memoria en muchos micros es de 64 k, en total, 65.536 octetos. En ocasiones oiremos hablar de semioctetos *(nibbles),* que es un grupo de cuatro bits, o sea, medio octeto, pero no es un término que se emplee demasiado. Hasta ahora hemos hablado de la manipulación de números que realizan los ordenadores como si esta fuera su única posibilidad, pero sabemos que también manipulan caracteres, y son capaces por tanto de escribir, almacenar y manejar textos.

## DECIMAL Y HEXADECIMAL

Nuestro sistema de numeración habitual es el decimal, basado en el 10. Utiliza los símbolos 0 al 9 y, para cantidades mayores que 9, los cambiamos de posición. Véase el ejemplo de abajo, con el número 3.057. Pueden también usarse otros sistemas. El hexadecimal tiene por base el 16. Los números 10, 11, 12, 13, 14 y 15 se representan con las letras A, B, C, D, E y F. En hexa se cuenta igual que en decimal. Se incrementa la cifra de la columna de las unidades hasta que llega a F; sumarle uno significa aumentar en 1 la cantidad de la siguiente columna (la del número multiplicado por 16) y poner a 0 la de las unidades.

Decimal

| ×1000 | ×100 | ×10 | ×1 |
|---|---|---|---|
| 3 | 0 | 5 | 7 |
| Tres miles | Cero cientos | Cinco decenas | Siete unidades |
| 3.000 | 0 | 50 | 7 |

= 3.057

Hexadecimal

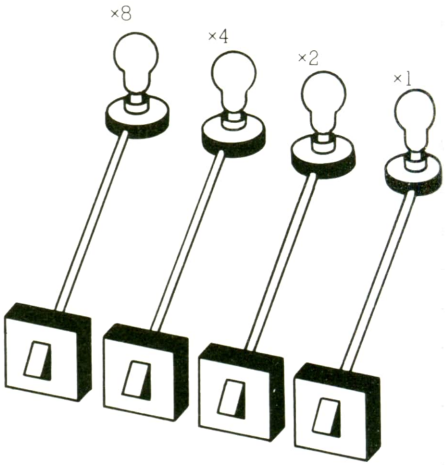
| ×4096 | ×256 | ×16 | ×1 |
|---|---|---|---|
| | | | 0 a 15 |

### El código ASCII

Así pues, ¿cómo se representan las letras en binario? Sencillamente, asignando a cada una de ellas un número en binario. Para poder intercambiar información entre distintos ordenadores, se ha adoptado un código común, el American Standard Code for Information Interchange o ASCII (pronúnciese «asqui»). Como puede verse en la tabla de las págs. 126-127, hay códigos ASCII para las le-

## BINARIO

La notación binaria es un sistema de numeración de base 2, y con él puede representarse cualquier número. Usa dos símbolos, el 0 y el 1, que en el ordenador son la presencia o ausencia de una corriente eléctrica. En binario, la única diferencia con el decimal o el hexa es que al ser sólo dos símbolos hacen falta más columnas para representar los mismos números. Las columnas en binario serán: ×1, ×2, ×4, ×8, ×16, ×32, ×64, etc. Así que, empezando con 0 y sumando 1 cada vez, los primeros números en binario serán (entre paréntesis los equivalentes decimales): 1 (1), 10 (2), 11 (3), 100 (4), 101 (5), 110 (6).

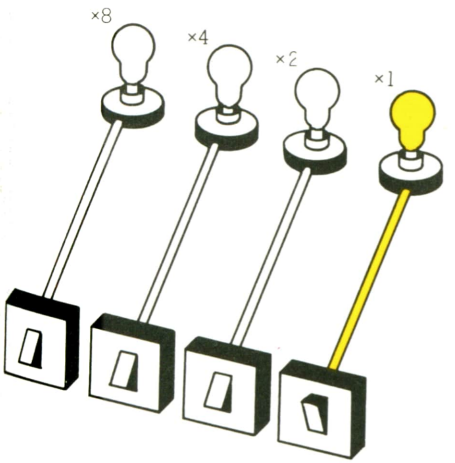



Ilustramos aquí el uso de una corriente eléctrica para contar en binario. Lámpara apagada significa cero, y encendida uno

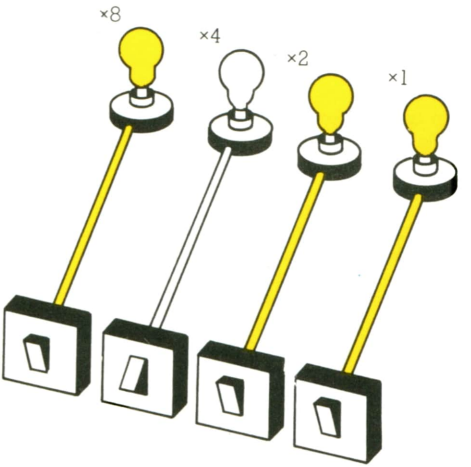
| ×32 | ×16 | ×8 | ×4 | ×2 | ×1 |
|---|---|---|---|---|---|
| | | | | | |

tras mayúsculas y minúsculas, signos de puntuación y diversos símbolos y números.

Dos preguntas que suele hacer el que da por vez primera con el ASCII son: ¿cómo distingue el ordenador entre los caracteres ASCII y los números binarios?, ¿por qué no se usa el octavo bit de cada octeto ASCII, con lo que se podría represen-




Encendiendo la lámpara de la columna de las unidades y manteniendo apagadas las demás se representa el 1.




El lector debe deducir, leyendo las lámparas de derecha a izquierda, que el número representado es el 11.

| Decimal | Equivalente binario | Decimal | Equivalente binario |
|---|---|---|---|
| 0 | 0 | 9 | 1001 |
| 1 | 1 | 10 | 1010 |
| 2 | 10 | 11 | 1011 |
| 3 | 11 | 12 | 1100 |
| 4 | 100 | 13 | 1101 |
| 5 | 101 | 14 | 1110 |
| 6 | 110 | 15 | 1111 |
| 7 | 111 | 16 | 10000 |
| 8 | 1000 | | |

tar doble número de caracteres? La respuesta a la primera es que el ordenador no sabe distinguirlos, eso es cosa del programador, que habrá que procurar que el ordenador no vaya a sumar «234» y «Z». En cuanto a la segunda, algunos micros usan el octavo bit para representar otro juego de caracteres, normalmente de tipo gráfico.

## DE BINARIO A HEXA

En binario resulta difícil buscar una discrepancia en una masa de números. Si dividimos cada octeto en dos semioctetos,el primer octero de la lista quedaría 1101 y 0001. En un semiocteto se pueden representar las cantidades del 0 al 15, y como en hexa esos mismos números se pueden expresar con un solo dígito, un octeto puede escribirse con dos dígitos (columnas de la derecha). Cualquier discrepancia resulta entonces obvia.

| | | | |
|---|---|---|---|
| **11010001** | **11010001** | **D1** | **D1** |
| **11010011** | **11010011** | **D3** | **D3** |
| **01110110** | **01110110** | **76** | **76** |
| **11001010** | **11001010** | **CA** | **CA** |
| **10001010** | **10011010** | **8A** | **9A** |
| **00111110** | **00111110** | **3E** | **3E** |
| **10010101** | **10010101** | **95** | **95** |
| **11000101** | **11000101** | **C5** | **C5** |

## LA TABLA ASCII

Los primeros 32 códigos ASCII son en su mayoría caracteres no imprimibles Casi todos se emplean para comunicaciones, pero algunos tienen aplicaciones generales. Así, CR significa retorno de carro y lleva el «cursor» de la pantalla al comienzo de la línea. HT tabula hacia la derecha, LF lleva el cursor a la línea de abajo y FF borra toda la pantalla. Muchos de estos caracteres se pueden teclear pulsando a la vez la tecla «control» y la letra adecuada, dando así lugar al «código de control». ASCII no tiene más símbolos para monedas que el $ y los ordenadores no norteamericanos lo «traducen».

| Código ASCII | Hexa | Carácter | Explicación |
|---|---|---|---|
| 007 | 07 | BEL ^G | Cuando se envía este código a la terminal suena la «campanilla». |
| 008 | 08 | BS ^H | El código de retroceso hace retroceder el cursor en pantalla un espacio. |
| 009 | 09 | HT ^I | La señal de tabulación horizontal funciona como el tabulador de una máquina de escribir: el cursor avanza hasta la siguiente tabulación fijada. |
| 010 | 0A | LF ^J | La señal de salto de línea avanza el papel 1 línea. En un monitor, sube la pantalla una línea. |
| 012 | 0C | FF ^L | El salto de página, en una impresora, hace avanzar el papel al comienzo de una nueva hoja. En muchos monitores borra la pantalla y lleva el cursor a la esquina superior izquierda. |
| 013 | 0D | CR ^M | El retorno de carro lleva la cabeza de impresión al comienzo de una nueva línea. |
| 027 | 1B | ESC ^[ | El código de «escape» se usa para detener pronto la ejecución del programa. |
| 127 | 7F | DEL | El código de borrado *(delete)* suele generarse con tecla especial, y borra el último carácter tecleado. |

| Código ASCII | Hexa | Carácter | Código ASCII | Hexa | Carácter | Código ASCII | Hexa | Carácter | Código ASCII | Hexa | Carácter |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 000 | 00 | NUL ^@ | 032 | 20 | (space) | 064 | 40 | @ | 096 | 60 | ` |
| 001 | 01 | SOH ^A | 033 | 21 | ! | 065 | 41 | A | 097 | 61 | a |
| 002 | 02 | STX ^B | 034 | 22 | " | 066 | 42 | B | 098 | 62 | b |
| 003 | 03 | ETX ^C | 035 | 23 | # | 067 | 43 | C | 099 | 63 | c |
| 004 | 04 | EOT ^D | 036 | 24 | $ | 068 | 44 | D | 100 | 64 | d |
| 005 | 05 | ENQ ^E | 037 | 25 | % | 069 | 45 | E | 101 | 65 | e |
| 006 | 06 | ACK ^F | 038 | 26 | & | 070 | 46 | F | 102 | 66 | f |
| 007 | 07 | BEL ^G | 039 | 27 | ' | 071 | 47 | G | 103 | 67 | g |
| 008 | 08 | BS ^H | 040 | 28 | ( | 072 | 48 | H | 104 | 68 | h |
| 009 | 09 | HT ^I | 041 | 29 | ) | 073 | 49 | I | 105 | 69 | i |
| 010 | 0A | LF ^J | 042 | 2A | * | 074 | 4A | J | 106 | 6A | j |
| 011 | 0B | VT ^K | 043 | 2B | + | 075 | 4B | K | 107 | 6B | k |
| 012 | 0C | FF ^L | 044 | 2C | , | 076 | 4C | L | 108 | 6C | l |
| 013 | 0D | CR ^M | 045 | 2D | - | 077 | 4D | M | 109 | 6D | m |
| 014 | 0E | SO ^N | 046 | 2E | . | 078 | 4E | N | 110 | 6E | n |
| 015 | 0F | SI ^O | 047 | 2F | / | 079 | 4F | O | 111 | 6F | o |
| 016 | 10 | DLE ^P | 048 | 30 | 0 | 080 | 50 | P | 112 | 70 | p |
| 017 | 11 | DC1 ^Q | 049 | 31 | 1 | 081 | 51 | Q | 113 | 71 | q |
| 018 | 12 | DC2 ^R | 050 | 32 | 2 | 082 | 52 | R | 114 | 72 | r |
| 019 | 13 | DC3 ^S | 051 | 33 | 3 | 083 | 53 | S | 115 | 73 | s |
| 020 | 14 | DC4 ^T | 052 | 34 | 4 | 084 | 54 | T | 116 | 74 | t |
| 021 | 15 | NAK ^U | 053 | 35 | 5 | 085 | 55 | U | 117 | 75 | u |
| 022 | 16 | SYN ^V | 054 | 36 | 6 | 086 | 56 | V | 118 | 76 | v |
| 023 | 17 | ETB ^W | 055 | 37 | 7 | 087 | 57 | W | 119 | 77 | w |
| 024 | 18 | CAN ^X | 056 | 38 | 8 | 088 | 58 | X | 120 | 78 | x |
| 025 | 19 | EM ^Y | 057 | 39 | 9 | 089 | 59 | Y | 121 | 79 | y |
| 026 | 1A | SUB ^Z | 058 | 3A | : | 090 | 5A | Z | 122 | 7A | z |
| 027 | 1B | ESC ^[ | 059 | 3B | ; | 091 | 5B | [ | 123 | 7B | { |
| 028 | 1C | FS ^/ | 060 | 3C | < | 092 | 5C | / | 124 | 7C | ¦ |
| 029 | 1D | GS ^] | 061 | 3D | = | 093 | 5D | ] | 125 | 7D | } |
| 030 | 1E | RS ^^ | 062 | 3E | > | 094 | 5E | ^ | 126 | 7E | ~ |
| 031 | 1F | US ^– | 063 | 3F | ? | 095 | 5F | — | 127 | 7F | (delete) |

# LA LOGICA INFORMATICA

Como vimos en el Módulo 13, para que un ordenador pueda manipular letras y números hace falta un sistema especial, el binario. Pero hasta ahora no hemos explicado qué significa exactamente «manipular» ni cómo lo hace el ordenador.

**La ALU**

Si repasamos un programa de ordenador (véase Módulo 27), comprobaremos que muchas instrucciones exigen que la máquina compare las cadenas binarias de información que constituyen los caracteres introducidos desde el teclado. Por ejemplo, una parte de un programa podría encargarse de analizar si una serie de fechas que los usuarios del sistema introducen son correctas. Para verificar algo tan sencillo hay que decirle al ordenador que compare la longitud de la cadena introducida para cerciorarse de que tiene el mismo número de octetos predeterminado por el programador. Luego habrá de comprobar que los signos de puntuación que figuren en la fecha están en la posición debida, comparando partes de la fecha con un modelo presente ya en el programa. Después, dependiendo también del programa concreto, si los dos últimos caracteres (que representan el año) caen dentro de ciertos límites, habrá que desviar el dato a una determinada posi-

ción de memoria y sumarlo o restarlo con la información allí almacenada previamente.

La parte del procesador que efectúa comparaciones se llama Unidad Aritmética y Lógica (ALU o *Arithmetic and Logic Unit).* Durante la ejecución del programa, se llevan los datos de la memoria a la ALU, donde son manipulados y luego devueltos a una posición de memoria o escritos por pantalla o impresora.

**Las tablas de verdad**

Para entender cómo realiza la ALU estas operaciones lógicas es preciso ver su mecanismo de funcionamiento. Supongamos que tenemos sólo dos bits, A y B, y que queremos saber si ambos son «1». Lo bueno sería poder realizar con ellos alguna operación que, sin alterarlos, diera por resultado un tercer bit (un «1» si ambos son «1» y un «0» si no lo son).En la terminología de la lógica esto se llama una tabla de verdad (véase figura del recuadro de la página siguiente).Esta operación concreta se llama AND: si A y B valen 1, el resultado es 1, si no, es 0. Supongamos, sin embargo, que queremos que el resultado de la operación sea 0 en lugar de 1. La tabla de verdad será diferente (véase dicha tabla) y resultará justamente lo contrario de AND, es decir, NOT AND o, abreviadamente, NAND. Son igualmente posibles otros tipos de operaciones lógicas. Por ejemplo, si queremos saber si A es 1, o lo es B, o lo son ambos, podemos construir una tabla de verdad como la de la página siguiente. Esta función se llama OR. Si A o B o ambos valen 1, el resultado es 1, si no, es 0. OR, lo mismo que AND, tiene un opuesto, que es NOR. Por último, puede resultar interesante saber si A o B valen 1, y a esta función se la llama OR excluyente o, en abreviatura, EOR o XOR.

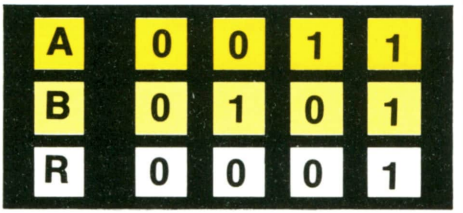
**Aplicación de las funciones lógicas**

Aplicando las funciones lógicas, el ordenador es capaz de crear nuevos bits de información. Si to-

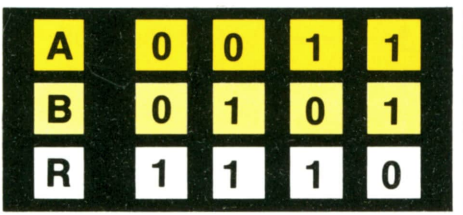
## TABLAS DE VERDAD

Las tablas de verdad son una visualización de operaciones lógicas para ver cómo trabaja un ordenador. En cada tabla de abajo realizamos una operación lógica sobre dos dígitos binarios (bits) A y B, para todas las posibles combinaciones de éstos. R es el resultado.

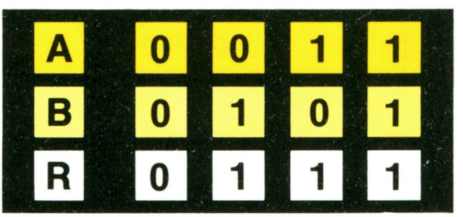
**AND.** La operación de AND nos permite comprobar si los dos bits son 1. El resultado es 1 si, y sólo si, tanto A como B son 1.



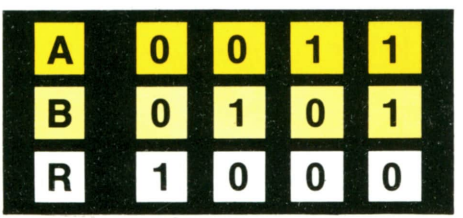

**NAND.** La operación lógica NAND es igual a la de AND pero resultado opuesto: un 0 cuando ambos operandos son 1.



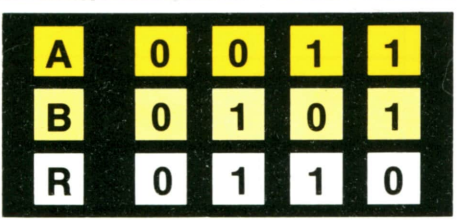

**OR.** La función OR indica que alguno de los operandos, o los dos, vale 1, dando por resultado 1. Si ambos son 0, el resultado será 0.




**NOR.** Igual que NAND es lo contrario de AND, NOR es lo contrario de OR. Da un 0 si alguno de los operandos, o ambos, vale 1.




**XOR.** La operación de OR excluyente o XOR produce un 1 si sólo uno de los operandos vale 1, y un 0 si son iguales los dos.



mamos el número binario 01100001 (61 hexadecimal o la «a» en ASCII) y lo sometemos a la operación AND con el 11011111, el resultado es:

| | |
|---|---|
| | **01100001** |
| **AND** | **11011111** |
| **Resultado** | **01000001** |

Si buscamos el resultado en la tabla ASCII (véase Módulo 13), comprobaremos que 01000001 es 41 hexa o la «A» ASCII. El resultado de esta operación es convertir un carácter ASCII en minúscula a su equivalente en mayúscula. Otro ejemplo, esta vez del uso de la función XOR:

| | |
|---|---|
| | **01000001** |
| **XOR** | **00100000** |
| **Resultado** | **01100001** |

El resultado de esta función es cambiar una «A» mayúscula por su equivalente minúscula.

Para llevar a cabo las operaciones lógicas de los dos ejemplos precedentes hay que ir avanzando bit a bit, de izquierda a derecha. Imaginando que el bit de arriba es A y el de abajo B, consultamos la tabla de verdad pertinente y averiguamos lo que sucede con cada combinación de bits concreta.

Para detectar discrepancias entre dos grupos de números, el ordenador puede realizar operaciones lógicas sobre cada octeto resultante para determinar si es cero o no:

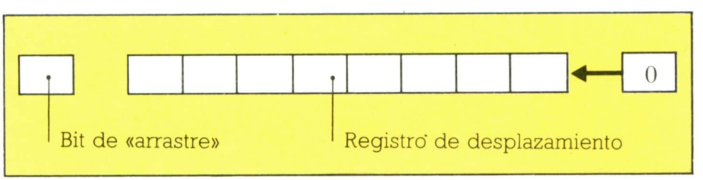
| | | | | |
|---|---|---|---|---|
| **11010001** | **XOR** | **11010001** | **da** | **00000000** |
| **11010011** | **XOR** | **11010011** | **da** | **00000000** |
| **01110110** | **XOR** | **01110110** | **da** | **00000000** |
| **10001010** | **XOR** | **10011010** | **da** | **00010000** |
| **11000101** | **XOR** | **11000101** | **da** | **00000000** |

Sometiendo a las diversas parejas de números a la operación XOR se pone de manifiesto inmediatamente la discrepancia en la línea 4.

### El desplazamiento lógico

Una operación que pudiera haberse utilizado para dar con la discrepancia del ejemplo anterior es la de «desplazamiento lógico». Imaginemos que tras completar esta operación tenemos nuestro octeto resultante en un registro especial que tiene un bit de más en uno de los extremos. Si conseguimos «introducir» un bit en el extremo opuesto (un cero en realidad) y «correr» los otros ocho un espacio, el bit que rebosase iría a parar a esa posición suplementaria, que luego podría examinarse para ver si este bit «de arrastre» era cero (véase figura de más abajo). Repitiendo esta operación ocho veces se pueden analizar los ocho bits del octeto y averiguar si el resultado es cero.

Aunque las funciones lógicas de este tipo parezcan complicadas, en realidad no consisten más que en realizar diversas operaciones sencillas a gran velocidad. Y esto forma parte de la «magia» del ordenador, la sorprendente velocidad con que ejecutan tareas en apariencia complicadas.



# LA LOGICA ELECTRONICA

Como ya sabemos, los ordenadores almacenan números y texto usando notación binaria. Utilizando ciertas funciones lógicas, tales como AND, OR y XOR (véase Módulo 14), el ordenador puede manipular o cambiar de posición los octetos. Pero para llevar a cabo dichas funciones se necesitan circuitos especiales, uno por cada operación.

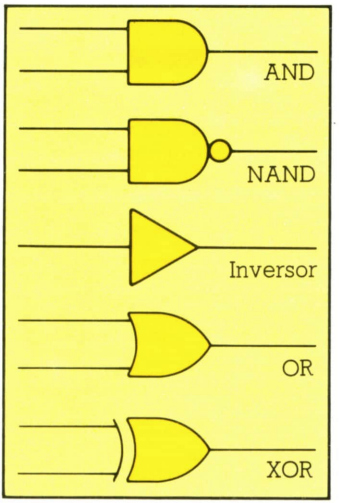
## De los relés a los *chips*

Cuando presentamos la notación binaria (véase Módulo 13), quedó claro que los números binarios se podían representar sencillamente con luces e interruptores. Pero operar así resultaría tedioso y exigiría múltiples cambios de interruptores. Sería mucho más fácil disponer de un único interruptor manual, un pulsador por ejemplo, y que el resto de las lámparas se encendiesen o apagasen automáticamente. Lo único que hace falta es un interruptor electrónico.

Uno o dos de los primeros ordenadores utilizaban relés electromecánicos para conseguir este efecto. Pero los relés suelen ser inestables, consumen mucha electricidad y son bastante lentos. Después se utilizaron interruptores electrónicos, primero válvulas y luego transistores. Se puede conseguir que cierto tipo de transistores funcione de esta manera, y son éstos los que se producen hoy en forma de circuitos integrados o *chips,* que

## SIMBOLOS LOGICOS ELECTRONICOS

Los ordenadores se construyen a partir de complicados agrupamientos de puertas lógicas. Para ganar tiempo y claridad, los diseñadores representan cada puerta por un símbolo. Los componentes más complejos carecen de símbolos. Sólo se escribe la función y se recuadra.



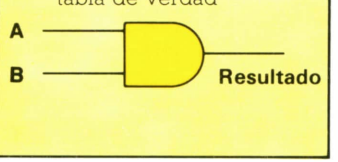

constan de gran número de transistores en redes complejas que llevan a cabo las funciones lógicas que vimos en el módulo anterior.

### Puertas lógicas

Los diseñadores de sistemas no tienen que preocuparse de cómo están conectados todos estos transistores; eso es cosa del fabricante del *chip*. Al diseñador sólo le preocupan las funciones lógicas que cada *chip* proporciona. Y estas funciones, o «puertas lógicas», están representadas por una serie de símbolos, representados en el recuadro que aparece sobre estas líneas.

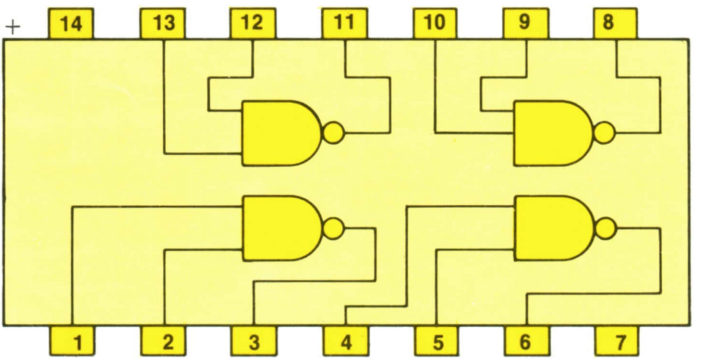
| | | | | |
|---|---|---|---|---|
| A | 0 | 0 | 1 | 1 |
| B | 0 | 1 | 0 | 1 |
| R | 0 | 0 | 0 | 1 |

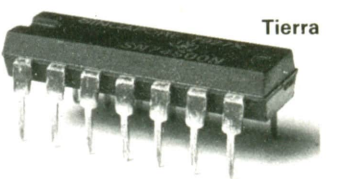
El circuito AND y su tabla de verdad



## ESQUEMA DE NAND

En comparación con el *chip* de un microprocesador, las puertas lógicas son dispositivos muy sencillos, por lo que los fabricantes suelen incluir varias en un mismo *chip*. El presente diagrama muestra la conexión de las puertas a las patas del *chip*.

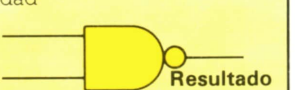



**El *chip* NAND.** Este es el *chip* real representado en el diagrama. Contiene 6 puertas NAND, cada una con 2 entradas y salidas. Las otras 2 patas son para alimentación y tierra.

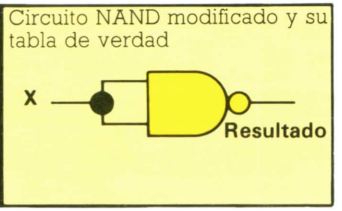
| | | | | |
|---|---|---|---|---|
| A | 0 | 0 | 1 | 1 |
| B | 0 | 1 | 0 | 1 |
| R | 0 | 1 | 1 | 0 |

El circuito NAND y su tabla de verdad

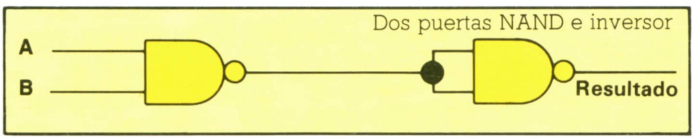



La puerta NAND es una de las funciones más útiles, y a partir de ella se pueden construir el resto de las funciones lógicas. Por ejemplo, si aplicamos *la misma* señal a las dos entradas de una puerta NAND se obtiene el siguiente circuito:

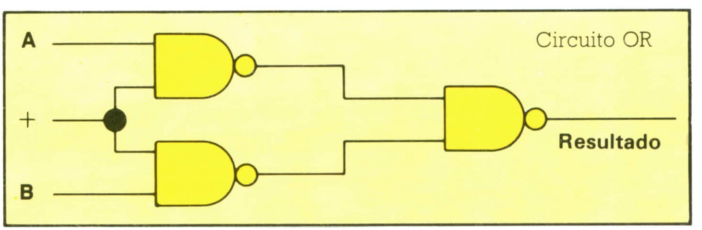
| X | 1 | 0 |
|---|---|---|
| R | 0 | 1 |



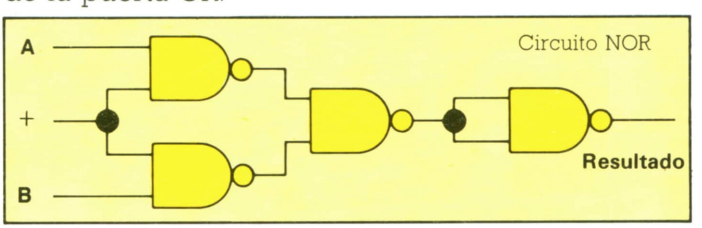

La puerta NAND se ha convertido en un «inversor» y con dos puertas NAND puede construirse una AND.



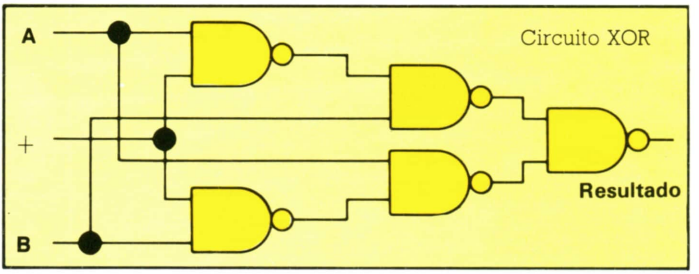

Una puerta OR se consigue empalmando tres puertas NAND, con tal de que en las dos primeras una de las entradas esté conectada a una fuente de voltaje positivo, para forzarlas al estado lógico «1».




Si seguimos adelante, usando la tabla de verdad del NAND y comparando los resultados con la tabla de verdad del OR que figura en el módulo anterior, comprobaremos que para crear una puerta NOR lo único que hace falta es invertir la salida de la puerta OR.



La circuitería para el OR excluyente o XOR es probablemente la más complicada.




Sin embargo, no es preciso echar mano de un montón de puertas NAND cada vez que se quiera representar una de las otras funciones lógicas. Existen *chips* especiales que dan el OR y el XOR, y se representan mediante símbolos especiales.

| A | 0 | 0 | 1 | 1 |
|---|---|---|---|---|
| B | 0 | 1 | 0 | 1 |
| R | 0 | 1 | 1 | 1 |

Símbolo y tabla de verdad de OR

| A | 0 | 0 | 1 | 1 |
|---|---|---|---|---|
| B | 0 | 1 | 0 | 1 |
| R | 1 | 1 | 1 | 0 |

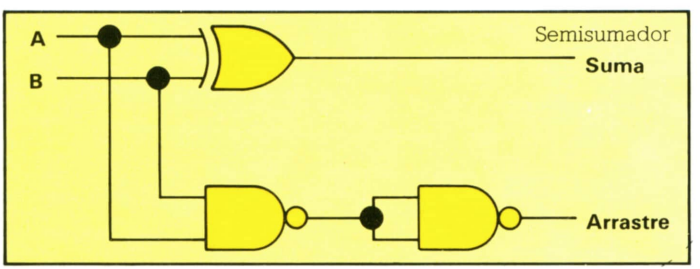
Símbolo y tabla de verdad de XOR

Como casi siempre se necesitan varias puertas lógicas, y como sale más barato poner unas cuantas en el mismo *chip,* normalmente se venden *chips* de 14 ó 16 patas. Los fabricantes publican manuales en los que se explica la disposición de las puertas dentro de todos sus *chips* mediante un diagrama esquemático.

**Uso de las puertas lógicas**

Excede del propósito de este libro dar una cumplida explicación de los aspectos prácticos de la construcción de estos circuitos. Lo que sí interesa es ver cómo se hacen experimentos sobre el papel. El circuito de abajo utiliza dos puertas NAND y una XOR, y se llama «semisumador».

Como puede verse, el circuito tiene dos entradas, A y B, pero también dos salidas. La puerta XOR funciona como es usual, dando un 1 si A o B, pero no ambos, son 1, y un 0 en cualquier otro caso. Pero de la primera puerta NAND saldrá un 0 si tanto A como B son 1 (y un 1 en caso contra-



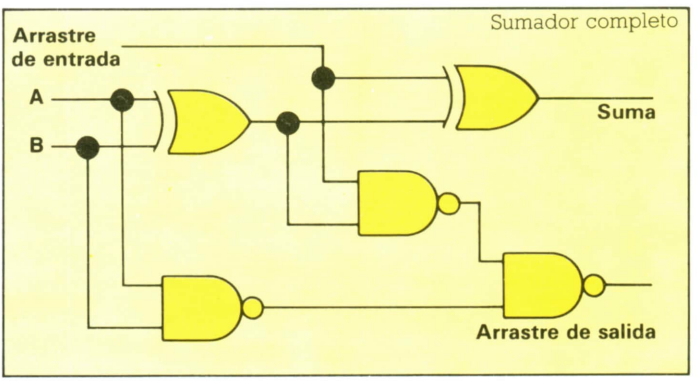

rio), que la segunda NAND invertirá, dando lugar a un 1. La tabla de verdad resultante será:

| A | 0 | 0 | 1 | 1 |
|---|---|---|---|---|
| B | 0 | 1 | 0 | 1 |
| Salida | 0 | 1 | 1 | 0 |
| Arrastre | 0 | 0 | 0 | 1 |

El circuito, en efecto, suma A y B. Entendiendo que la señal de salida es la columna de las unidades y la de rebosamiento la columna a su izquierda, lo que tenemos es la sencilla suma:

| | | |
|---|---|---|
| A | + | 1 |
| B | | 1 |
| Resultado | | 10 |

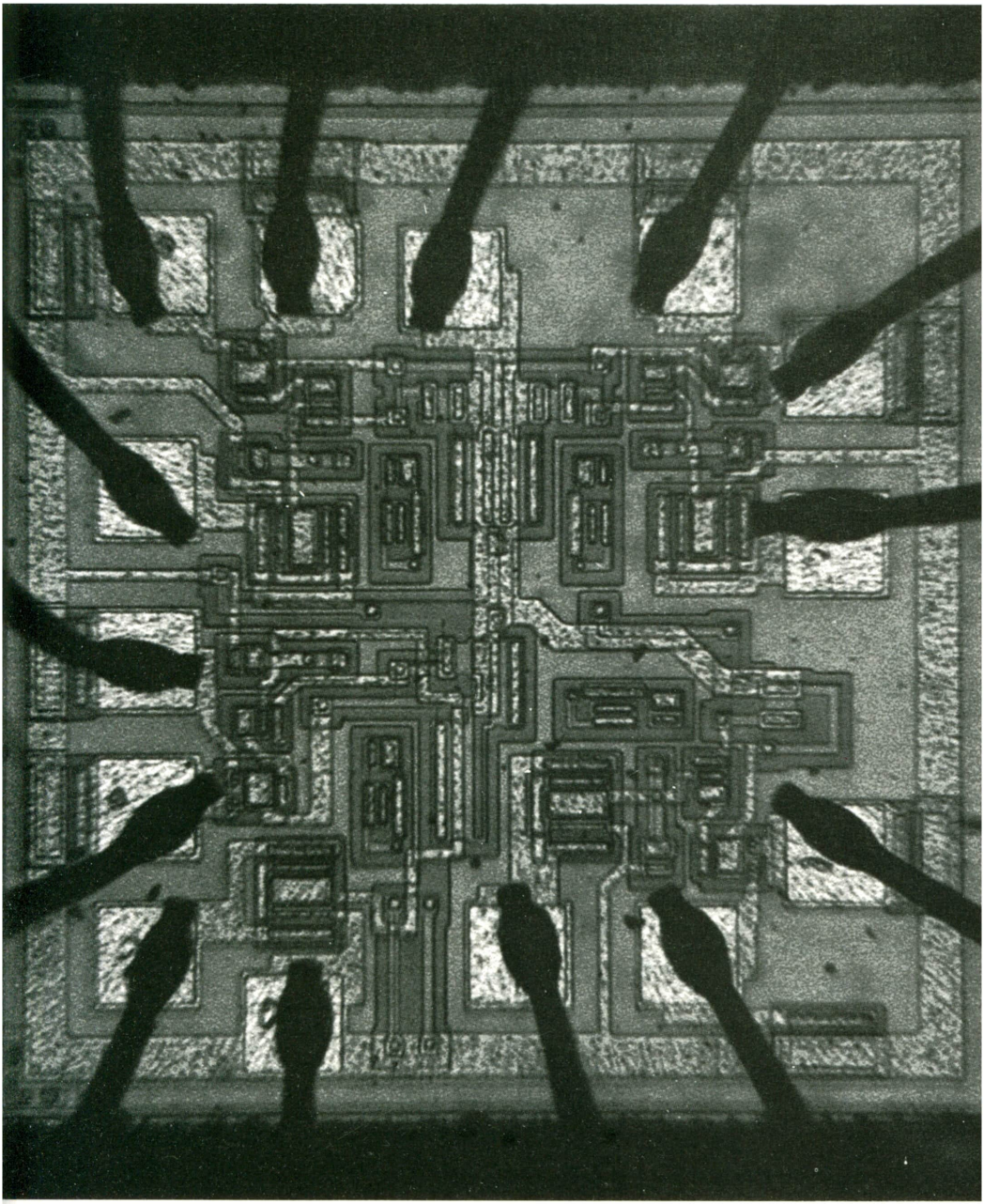
Este tipo de circuito se llama «semi» sumador porque, aunque genera un arrastre, no admite como entrada el arrastre generado por un semisumador anterior. Para sumar, teniendo en cuenta el arrastre (las que «nos llevamos») hace falta un circuito más complejo: el «sumador completo».

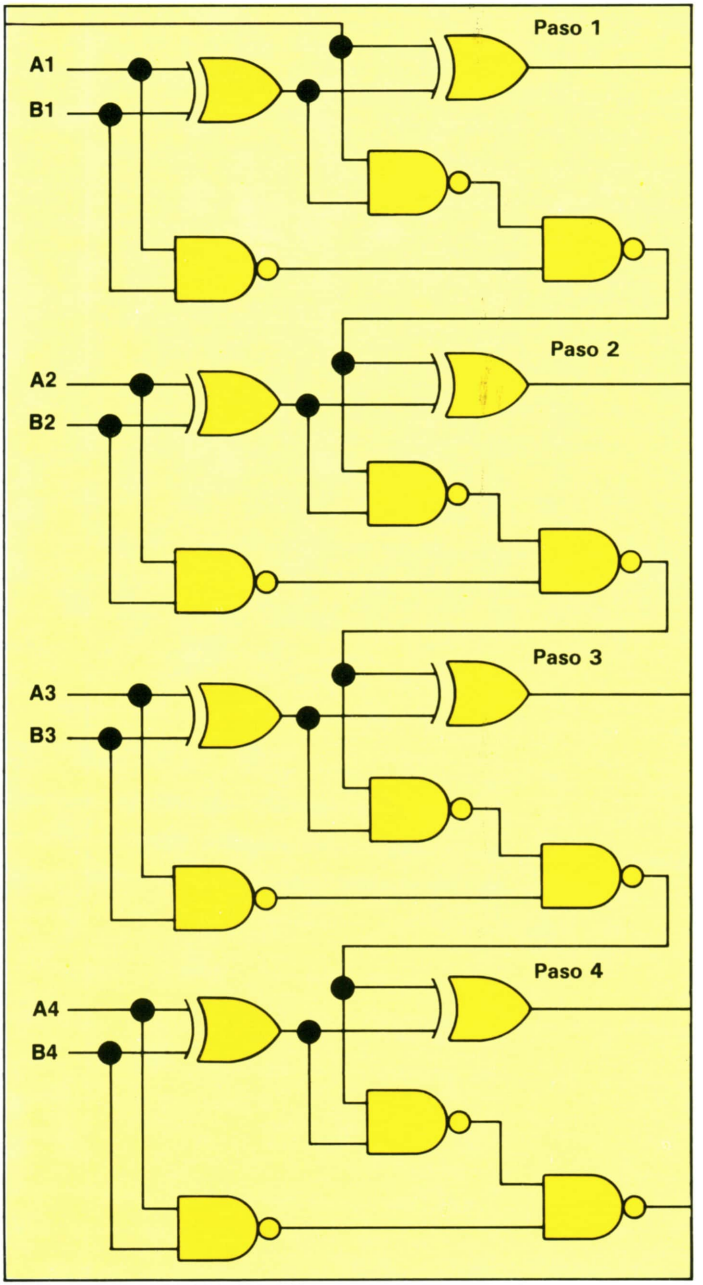



Recorriendo el circuito y consultando las tablas de verdad de las puertas NAND y XOR comprobaremos que este circuito da lugar a:

| A | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|
| B | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| A-ent. | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| Suma | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| A-sal. | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |

***Chip* de lógica.** Contemplado al microscopio, el *chip* no parece más que un amasijo de formas desprovistas de sentido. Sin embargo, lo cierto es que estas formas constituyen la estructura de silicio y otros elementos, construida capa a capa, que da lugar a los transistores y otros componentes. Interconectándolos se obtienen las puertas lógicas. El *chip* de la figura es muy simple; se trata de un grupo de cuatro puertas NAND de dos entradas. Las líneas oscuras que salen del *chip* son los hilos que lo conectan a las patas de su soporte exterior. Transportan las señales, la alimentación y la conexión a tierra. Un *chip* de microprocesador es mucho más complejo que éste (véase Módulo 17).




Paso 1
A1
B1
Paso 2
A2
B2
Paso 3
A3
B3
Paso 4
A4
B4

Los sumadores completos se pueden encadenar «en cascada», con lo que se consigue sumar números binarios enteros. En el diagrama de la izquierda aparecen cuatro de estos sumadores, cada uno atiende a un bit, por lo que con esta configuración sólo pueden manejarse simultáneamente semioctetos (cuatro bits). Para sumar dos octetos precisaríamos otros tantos. Como todos son sumadores completos, al final de la cascada queda siempre una señal de arrastre de salida, que se puede hacer pasar por circuitos especiales que nos digan si el resultado de sumar los dos números es demasiado grande y no cabe en un octeto. Los números de este tamaño habrá que repartirlos entre dos o incluso más octetos.

# EL ORDENADOR REAL

Hemos comenzado a darnos cuenta de que las puertas lógicas pueden llevar a cabo operaciones muy diversas con los datos y determinar el contenido del dato resultante (véanse Módulos 14 y 15). El siguiente paso es aplicar todo esto.

**Basculadores *(flip-flops)***

El elemento básico de todos los circuitos lógicos es la puerta NAND. Sería posible construir todo un ordenador sin usar más que puertas NAND pero, por comodidad y economía, se fabrican muchos más *chips* especializados que pueden utilizarse en su lugar. Por ejemplo, los basculadores (véase figura 1). Resultan de interés porque permiten almacenar un bit que luego puede recuperarse. Poniendo ocho en paralelo se podrá almacenar un octeto. El basculador es, en realidad, una memoria, y la RAM de un ordenador no es más que una inmensa colección de basculadores.

La figura 2 detalla la construcción de una celda de memoria de un octeto. Colocando datos en las líneas de entrada del 1 al 8 y luego abriendo la línea del reloj para enviar un «1» durante un breve período de tiempo, puede cargarse el octeto en la celda de memoria. Este método tiene, sin embargo, una pega, aparte de que en el circuito mostrado sólo se pueda mantener un octeto. Cuando cargamos el valor en memoria, aparece inmedia-

tamente como salida, lo que no conviene a menos que vayamos a usarlo de inmediato. Lo que necesitamos, así, es un modo de separar las salidas del resto del sistema hasta el momento en que queramos leer el contenido de la celda de memoria.

Este problema puede resolverse usando como interruptor una puerta AND. Poniendo una puerta AND a la salida del basculador, basta enviar una señal de «lectura» a la otra entrada de la puerta AND y ya tenemos un medio de retener la salida mientras haga falta. Aparece ilustrado en la figura 3. La celda de memoria se puede controlar con dos señales, una que permita escribir datos en ella (el control de escritura) y otra para leerlos. Se pone así de manifiesto una importante propiedad del funcionamiento de la memoria: el contenido de una celda de memoria sólo puede alterarlo una operación de escritura. Cuando la señal de lectura permite la salida del contenido de la celda al *bus* de datos, lo que se recibe es una *copia* de dicho contenido, sin que éste se vea afectado.

Una memoria en la que sólo se pueda almacenar un octeto no resulta muy práctica. Los ordenadores modernos necesitan miles —millones a veces— de celdas de memoria. Es preciso pues contar con algún medio de activar y desactivar individualmente las celdas tanto para las operaciones de lectura como para las de escritura. Puede conseguirse con puertas AND; en la figura 4 se presenta un sencillo interruptor de tres vías. La clave del sistema está en el *bus* de direcciones, en este caso de dos hilos, por donde se puede enviar el número binario correspondiente al dispositivo que se desea activar. Con dos líneas pueden formarse cuatro combinaciones binarias: 00, 01, 10 y 11. Si se reserva la señal 00 para indicar que no se alude a ninguno, con este esquema se pueden controlar hasta tres dispositivos. En las entradas de dos de las puertas AND se ponen unos inversores para garantizar que sólo una de las puertas reciba la señal 11 que la activa.

Una vez ideado un método de «direccionar» in-

En un principio se construían cada una de las puertas lógicas de un ordenador con válvulas o transistores, pero hoy los *chips* incorporan centenares de puertas básicas. Se puede predecir la salida para todas las posibles combinaciones de entradas, pues obedece a reglas lógicas. Para montar funciones más complejas es preciso analizar las posibles entradas, determinar las salidas deseadas y enlazar puertas.

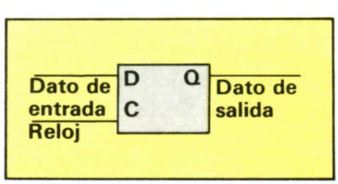



**Figura 1.** En un basculador, un pulso en la entrada del «reloj» cambia la salida a la condición de «dato de entrada»; cesado el pulso, la salida se mantendrá al nivel de la entrada.

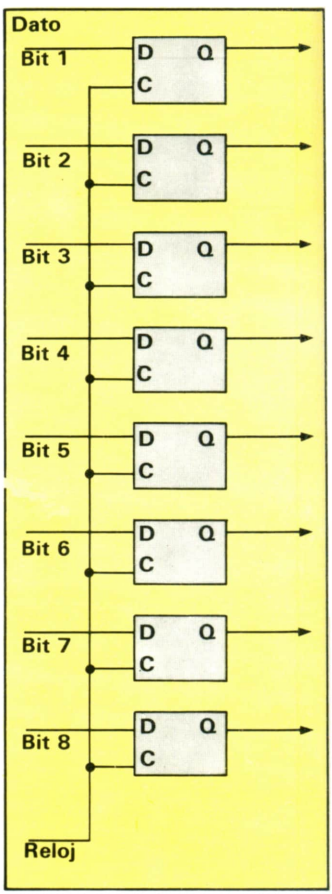



**Figura 2.** Con ocho basculadores en paralelo se puede almacenar un octeto. Para ello se aplica éste a las líneas de «entrada» y se manda un pulso por la del reloj.

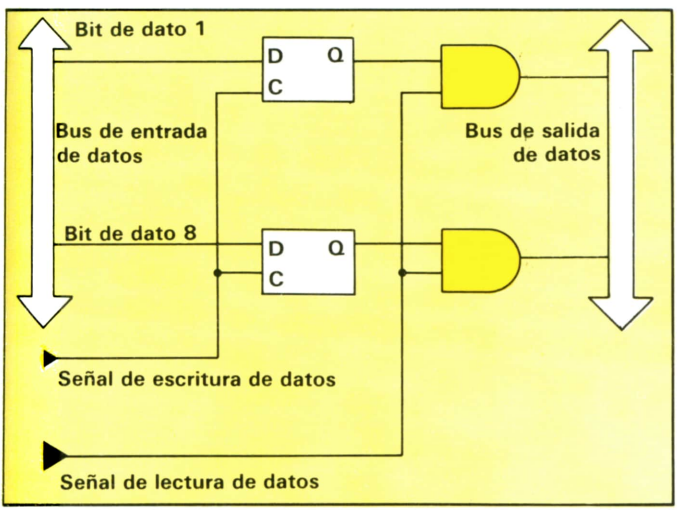
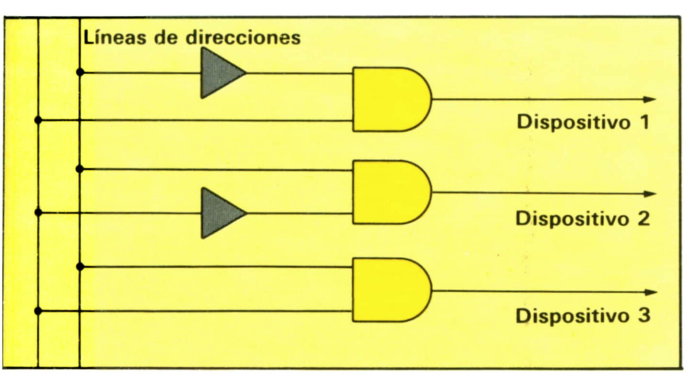
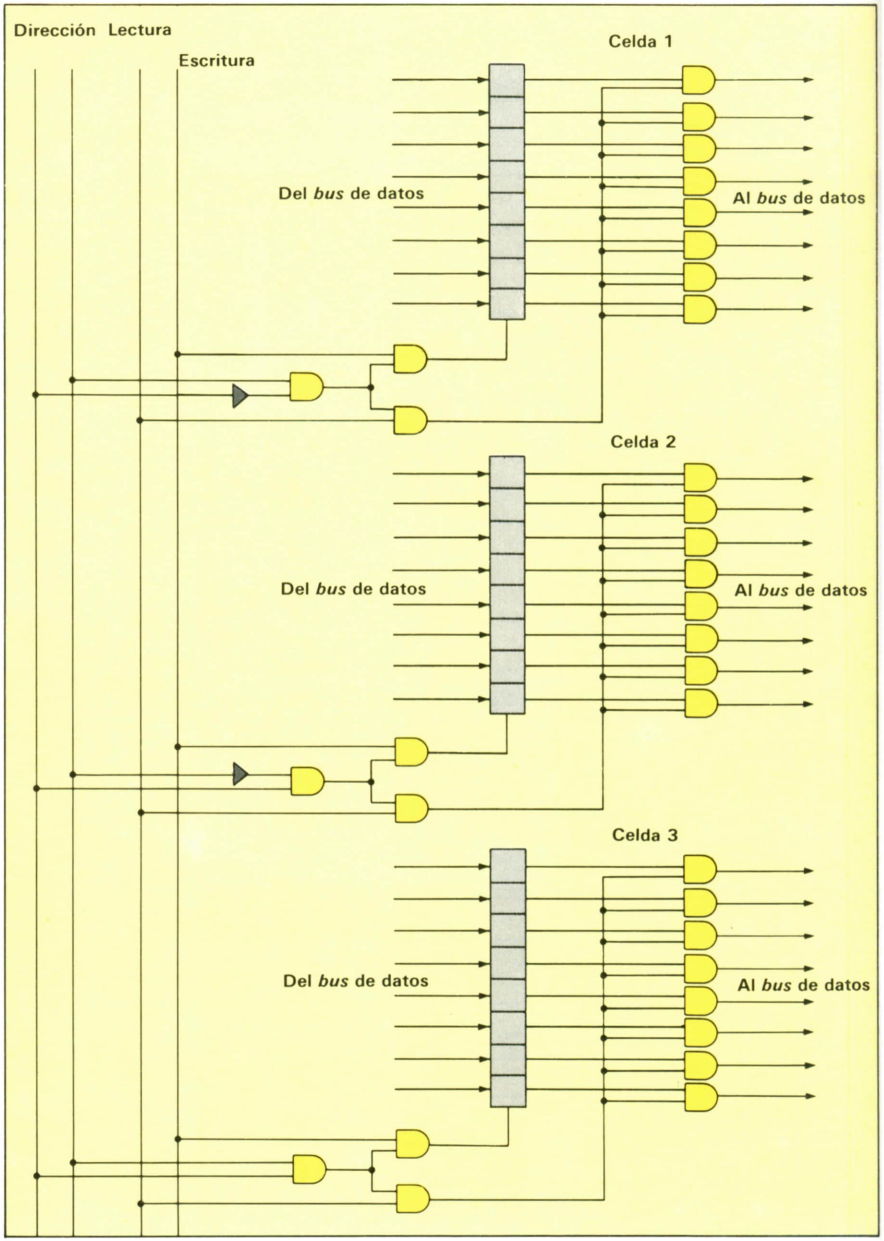



**Figura 3.** La salida del basculador constituye una de las entradas de la puerta AND, siendo la otra la señal de control. Activándola conseguiremos que la salida del AND refleje la procedente del basculador.

**Figura 4.** Combinando puertas AND e inversores se pueden decodificar señales. Si ambas líneas de control son 0,no se selecciona ningún dispositivo.



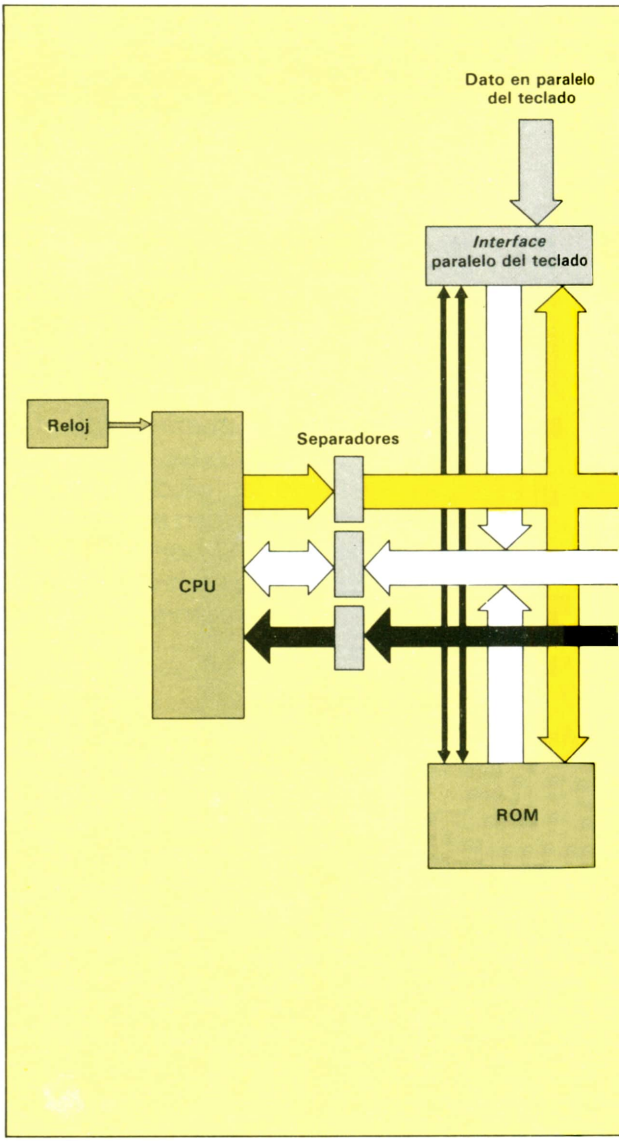

Dirección
Lectura
Escritura
Celda 1
Del *bus* de datos
Al *bus* de datos
Celda 2
Del *bus* de datos
Al *bus* de datos
Celda 3
Del *bus* de datos
Al *bus* de datos

dividualmente las celdas de memoria y de controlar tanto su escritura como su lectura, es posible construir una memoria operativa. La figura 5 presenta un sistema de memoria de tres octetos, describiendo cómo actúan las señales de dirección, lectura y escritura para que se puedan introducir o sacar datos en una posición de memoria concreta. En un ordenador, todas estas señales, lectura, escritura y dirección, las proporciona la CPU.

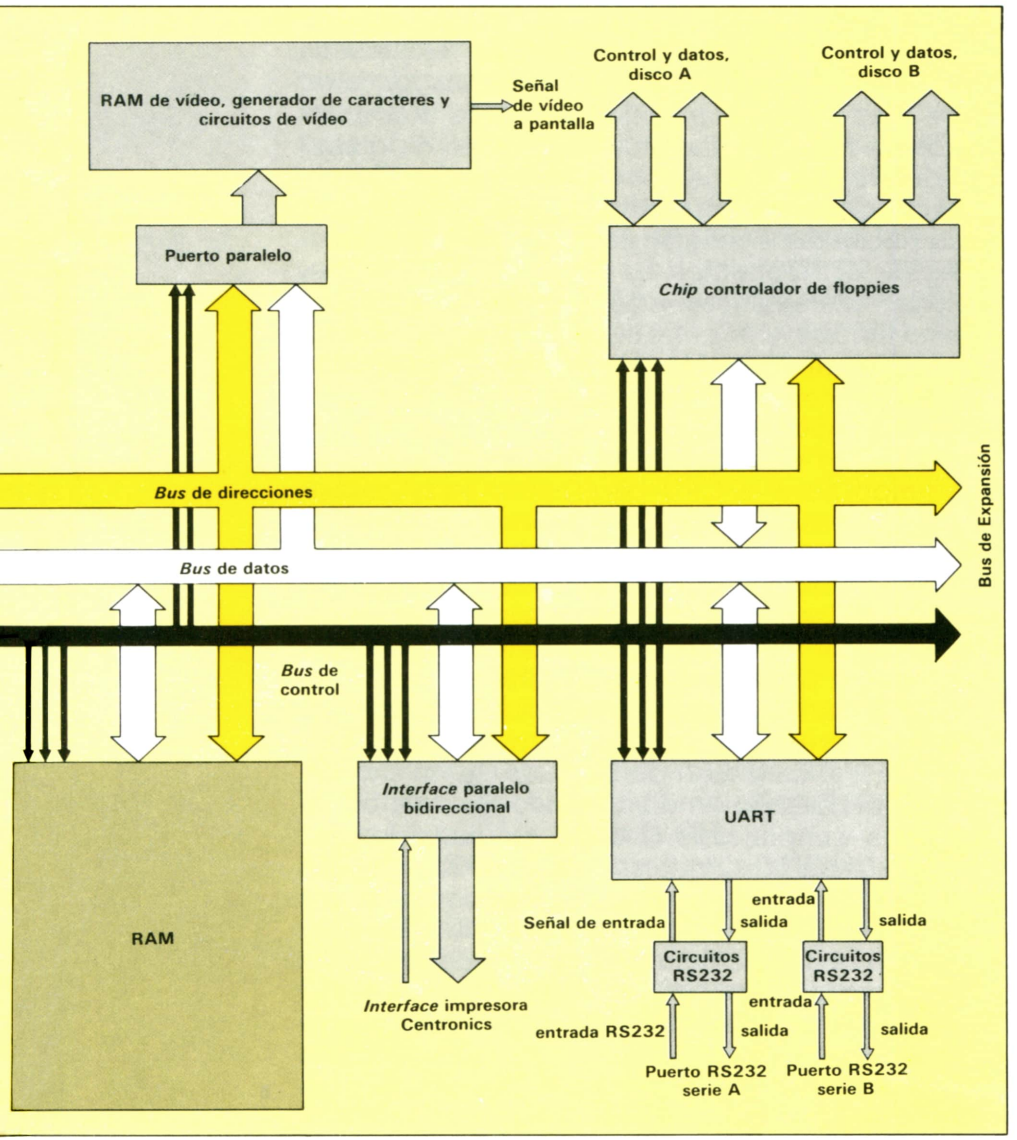
La CPU se rige por los mismos principios. Por ejemplo, los registros de la CPU no son más que celdas de memoria, de 8 o de 16 bits, que residen en el mismo *chip* de la CPU, pudiéndose por tanto acceder a su contenido con más rapidez que al de las celdas de la memoria principal. Los datos de estos registros internos se manejan dentro de la CPU por un sistema de puertas análogo al ilustrado en la figura 5. Además, tales puertas activan y desactivan las diversas partes de la unidad arit-

**Figura 5.** Combinando los principios bosquejados en los cuatro diagramas precedentes se obtiene un sencillo método de almacenar y recuperar datos a voluntad. Tenemos aquí tres celdas de memoria. Enviando una dirección binaria —01, 10 ó 11— por las líneas de direcciones seleccionamos una de las tres. Un 1 por la línea de escritura sirve de reloj para el grupo de basculadores seleccionado, copiando así los datos que en ese momento estén en el *bus* de datos. Seleccionando un grupo de basculadores y enviando luego un 1 por la línea de lectura, el contenido de la celda seleccionada pasará al *bus* de datos. Este es, pues, el fundamento de la memoria de un ordenador. La CPU puede escribir y leer datos en cualquier posición de memoria que se elija. La mayoría de los microordenadores tienen por lo menos 16 líneas de direcciones, lo que les permite acceder a miles de celdas de memoria. Un sistema microinformático típico consta de varios componentes básicos, interconectados por tres *buses* que transportan señales e información por todo el sistema. Como las señales de la CPU no tienen potencia suficiente para gobernar todos los componentes del sistema, han de ser amplificadas por los separadores. Las distintas unidades del sistema decodifican luego el *bus* de direcciones y las líneas de control con sus propios circuitos (que no aparecen en la figura) y sólo se activan cuando la CPU ha lanzado la combinación de señales adecuada.

**Figura 6.**
Representa este diagrama un sistema de gestión típico, montado en torno a dos discos flexibles. Al encenderlo, el *software* de la ROM trae del disco el sistema operativo, que toma el control de inmediato, quedando luego desactivada la ROM. El sistema operativo controla todas las funciones rutinarias de bajo nivel, por ejemplo, recoger los caracteres tecleados o enviar los códigos ASCII a la sección de vídeo para que salgan por pantalla. Muchos fabricantes hacen llegar las señales de dirección, dato y control a cierto conector, para que se puedan añadir al sistema otros componentes, normalmente tarjetas con más RAM o *interfaces* suplementarias.




mética y lógica de la CPU. La acción es compleja y está supervisada por la unidad de control de la CPU, que interpreta las instrucciones binarias que recibe, traduciéndolas en el conjunto de señales precisas para activar las diversas partes de la CPU




que requiere tal operación. Envía asimismo señales de activación a las puertas AND, permitiendo que los datos recorran estos circuitos saliendo y entrando de los registros o del canal de datos, según convenga.

**Un sistema informático basado en disco**

En los microordenadores modernos existe una tendencia a aumentar el número de funciones de cada *chip* y a crear *chips* cada vez más complejos. La industria ha llegado al punto de disponer de *chips* «inteligentes» para controlar discos, capaces de realizar operaciones que hace tan sólo unos pocos años exigían tarjetas enteras abarrotadas de componentes. La ejecución de las operaciones matemáticas más complejas, consume mucho tiempo, y por ello se han creado *chips* especiales para procesos matemáticos. La CPU envía dos números al *chip* matemático, junto con un código que indica la operación a que hay que someterlos y el *chip* realiza el cálculo deseado, quedando libre la CPU para hacer otras cosas.

Una vez resuelto qué CPU se va a usar, cuánta memoria se necesita y si harán falta recursos gráficos o de color, puertos de entrada y salida o unidades de disco, hoy diseñar un sistema equivale a seleccionar los controladores «inteligentes» precisos y, casi literalmente, conectarlos.

Para dar una idea de la diversidad de componentes que se reúnen en un micro, se muestra en la figura 6 el diagrama de bloques de un sistema de gestión basado en disco típico. Se trata de un diagrama simplificado. No se ha incluido, por ejemplo, toda la circuitería de la RAM y de los puertos de entrada-salida, y tampoco todas las fuentes de alimentación, pero con tantas líneas el diagrama resultaría demasiado difícil de entender.

En el corazón del sistema se encuentra la CPU. El reloj del sistema genera una corriente continua de pulsos gracias a la cual se realizan a un ritmo regular las operaciones de la CPU, en el orden correcto y a su debido tiempo. Otros componentes, en especial la circuitería de vídeo y el UART (véase Módulo 20) pueden estar también conectados a este reloj o contar con el suyo propio.

Hemos visto que la CPU genera la dirección de las posiciones de memoria o de los puertos de entrada-salida con que se quiere trabajar y que re-

cibe datos de estos componentes o se los envía. Estas direcciones se encarga de distribuirlas por todo el sistema el *bus* de direcciones, que consiste en una serie de conexiones (normalmente 16 para un sistema de 8 bits) a las que se enlazan los circuitos que decodifican las direcciones para cada componente. Existe análogamente un *bus* de datos (de 8 bits para un sistema de 8 bits) que trasvasa datos entre la CPU y el resto del sistema. La CPU genera y recibe asimismo varias señales de control, que transporta el *bus* de control. En la figura 6 se muestran sólo las básicas (lectura, escritura y otras dos que indican si la operación de lectura o escritura se refiere a la memoria o a la entrada-salida) en aras de la simplificación. Las señales que emanan de la CPU no son especialmente potentes; serán capaces de alimentar sólo un corto número de puertas lógicas, pero salvo en los sistemas más rudimentarios serán abundantes las puertas lógicas unidas a los *buses* de direcciones, datos y control. Así que lo primero que se hace con las señales que salen de la CPU es darles potencia suficiente para «manejar» todas las puertas lógicas que haga falta. Esta es la función de los tampones *(buffers)* que hay junto a la CPU: amplificar las señales para evitar la sobrecarga de la circuitería interna de la CPU.

Este sistema lleva una cierta cantidad de ROM que contiene un programa llamado «protocargador» *(bootstrap loader)*. Cuando se enciende el ordenador, este circuito se encarga de que la CPU tome sus primeras instrucciones de esta ROM. En un sistema típico, estas instrucciones prepararán unos cuantos procedimientos operativos básicos y luego cargarán en la RAM el sistema operativo, leyéndolo de alguna de las unidades de disco. Hecho esto, el cargador ordena a la CPU ejecutar los programas del sistema operativo en la RAM, y a partir de ese momento se prescinde de la ROM.

El resto del sistema consiste esencialmente en una serie de puertos de entrada-salida para unir el ordenador a una impresora o a un monitor, por

ejemplo. El más complejo de estos componentes es el controlador de disco, que para el sistema suele ser equivalente a varios puertos de entrada-salida, algunos para control y uno o dos para datos. También la *interface* de la pantalla puede equivaler a uno o varios puertos de E/S.

El teclado lleva su propio *chip* de control, que averigua qué tecla se ha pulsado y genera el código ASCII correspondiente. La CPU vigila el puerto de entrada-salida del teclado si su programa le dice que ha de esperar un dato del teclado y aguarda hasta que se haya pulsado un carácter. Luego, lo normal es que los códigos ASCII sean transferidos, uno a uno, a una zona reservada de memoria, hasta que el usuario teclea un retorno de carro para indicar el final del proceso.

Todas las máquinas de gestión y muchos modelos domésticos cuentan con la posibilidad de conectar al ordenador una impresora. Son dos los métodos más corrientes de conexión: las *interfaces* paralelo Centronics y serie RS232. Lo normal es que se disponga de ambas *interfaces* para que el usuario pueda emplear la que cuadre con su impresora. En el sistema ilustrado en la figura 6, la *interface* RS232 es proporcionada por el UART y sus circuitos asociados, que convierten la salida del UART a los niveles de voltaje RS232. Los UART más modernos suministran dos puertos independientes, con lo que el sistema cuenta con una segunda *interface* RS232 a la que se puede conectar, por ejemplo, un módem (véase Módulo 24).

# TÍTULOS PUBLICADOS

| | |
|---|---|
| 1 Carl Sagan | *La conexión cósmica* |
| 2 Isaac Asimov | *Introducción a la ciencia (I)* |
| 3 | *El futuro de la exploración del espácio** |
| 4 Isaac Asimov | *Introducción a la ciencia (II)* |
| 5 Bertrand Russell | *ABC de la relatividad* |
| 6 Isaac Asimov | *Fotosíntesis* |
| 7 C. Rayner | *La mente humana** |
| 8 Desmond Morris | *El mono desnudo* |
| 9 Alvin Toffler | *La tercera ola (I)* |
| 10 Alvin Toffler | *La tercera ola (II)* |
| 11 Richard Leakey | *La formación de la humanidad (I)** |
| 12 Werner Heisenberg | *La imagen de la naturaleza en la física actual* |
| 13 Pierre P. Grassé | *El hombre, ese dios en miniatura* |
| 14 David Dickson | *Tecnología alternativa* |
| 15 Richard Leakey | *La formación de la humanidad (II)** |
| 16 Stephen Jay Gould | *El pulgar del panda* |
| 17 Walter C. Patterson | *La energía nuclear* |
| 18 Erwin Schrödinger | *¿Qué es la vida?* |
| 19 C. Rayner | *El cuerpo humano(I)* |
| 20 Jacques Monod | *El azar y la necesidad* |
| 21 Stephan L. Chorover | *Del Génesis al genocidio* |
| 22 J. E. Lovelock | *Gaia, una nueva visión de la vida sobre la Tierra* |
| 23 C. Rayner | *El cuerpo humano (II)** |
| 24 Desmond Morris | *El zoo humano* |
| 25 K. Lorenz y otros | *Hombre y animal* |
| 26 Charles Sherrington | *Hombre* versus *Naturaleza* |
| 27 K. Gatland | *Exploración del espacio (II)** |
| 28 Robert Fouet/Charles Pomerol | *Las montañas* |
| 29 Paul Colinvaux | *Por qué son escasas las fieras* |
| 30 F. Jacob y otros | *Biología molecular* |
| 31 K. Gatland | *Exploración del espacio (III)** |
| 32 Alan Ross Anderson | *Controversia sobre mentes y máquinas* |
| 33 Pierre George | *El medio ambiente* |
| 34 Xavier Le Pichon/Guy Pautot | *El fondo de los océanos* |
| 35 K. Gatland | *Exploración del espacio (IV)** |
| 36 H. J. Eysenck | *Raza, inteligencia y educación* |
| 37 Fernand Moreau | *Alcaloides y plantas alcaloideas* |
| 38 Bruce A. Bolt | *Terremotos* |

| | |
|---|---|
| 39 R. Hardy, P. Wright, J. Gribbin, J. Kington | *El libro del clima (I)** |
| 40 Recopilación de artículos de La Recherche | *Astrofísica* |
| 41 Sydney P. Clark | *La estructura de la Tierra* |
| 42 Recopilación de artículos de La Recherche | *Las nuevas energías* |
| 43 P. Hardy, P. Wright, J. Gribbin, J. Kington | *El libro del clima (II)** |
| 44 H. J. Eysenck | *Experimentos en terapia de la conducta I. Inhibición recíproca* |
| 45 H. J. Eysenck | *Experimentos en terapia de la conducta II. Métodos de condicionamiento* |
| 46 H. J. Eysenck | *Experimentos en terapia de la conducta III. Experimentación con niños* |
| 47 R. Hardy, P. Wright, J. Gribbin, J. Kington | *El libro del clima (III)** |
| 48 Tom Logsdon | *Robots: una revolución* |
| 49 B. F. Skinner | *Sobre el conductismo* |
| 50 H. Takeuchi/S. Uyeda/H. Kanamori | *¿Qué es la Tierra?* |
| 51 Desmond Morris | *El hombre al desnudo (I)** |
| 52 Paul Chovin/André Roussel | *La polución atmosférica* |
| 53 Pierre Rousseau | *Astronomía sin telescopio* |
| 54 Félix Trombe | *Las aguas subterráneas* |
| 55 Desmond Morris | *El hombre al desnudo (II)** |
| 56 A. R. Luria | *El cerebro en acción (I)* |
| 57 A. R. Luria | *El cerebro en acción (II)* |
| 58 D. H. Tarling/M. P. Tarling | *Derivas continentales* |
| 59 Desmond Morris | *El hombre al desnudo (III)** |
| 60 Pierre Rousseau | *La luz* |
| 61 Stephen J. Gould | *La falsa medida del hombre* |
| 62 Jean-Jacques Matras | *El sonido* |
| 63 Desmond Morris | *El hombre al desnudo (IV)** |
| 64 Richard Feynman | *El carácter de la ley física* |
| 65 Michel Béguery | *La explotación de los océanos* |
| 66 Ivan P. Pavlov | *Actividad nerviosa superior* |

* *Volumen extra, en color.*